PENGAMANAN PENGIRIMAN PESAN DALAM GAMBAR MENGGUNAKAN STEGANOGRAFI DENGAN *DISCRETE COSINE TRANSFORM* (DCT) DAN KRIPTOGRAFI *RIVEST CODE* 6 (RC6)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk
Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

ENDRIKO MARTOFORI
10851002890

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2013

LEMBAR PERSETUJUAN

PENGAMANAN PENGIRIMAN PESAN DALAM GAMBAR MENGGUNAKAN STEGANOGRAFI DENGAN *DISCRETE COSINE TRANSFORM* (DCT) DAN KRIPTOGRAFI *RIVEST CODE* 6 (RC6)

TUGAS AKHIR

oleh:

ENDRIKO MARTOFORI
10851002890

Telah diperiksa dan disetujui sebagai laporan tugas akhir
di Pekanbaru, pada tanggal 11 Oktober 2013

Koordinator Tugas Akhir	Pembimbing
Muhammad Affandes, M.T	**Reski Mai Candra, S.T, M.Sc**
NIK. 130 510 030	**NIK. 130 150 032**

LEMBAR PENGESAHAN

PENGAMANAN PENGIRIMAN PESAN DALAM GAMBAR MENGGUNAKAN STEGANOGRAFI DENGAN *DISCRETE COSINE TRANSFORM* (DCT) DAN KRIPTOGRAFI *RIVEST CODE* 6 (RC6)

TUGAS AKHIR

Oleh :

ENDRIKO MARTOFORI
10851002890

Telah dipertahankan di depan sidang dewan penguji
Sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau
Di Pekanbaru, pada tanggal, 10 Oktober 2013

Pekanbaru, 11 Oktober 2013
Mengesahkan,

Dekan

Dra. Hj. Yenita Morena, M.Si
NIP. 19601125 198503 2 002

Ketua Jurusan

Elin Haerani, S.T, M.Kom
NIP. 19810523 200710 2 003

DEWAN PENGUJI

Ketua : Reski Mai Candra, S.T, M.Sc
Sekretaris : Reski Mai Candra, S.T, M.Sc
Penguji I : Lestari Handayani, S.T, M.Kom
Penguji II : Iwan Iskandar, M.T

PENGAMANAN PENGIRIMAN PESAN DALAM GAMBAR MENGGUNAKAN STEGANOGRAFI DENGAN *DISCRETE COSINE TRANSFORM* (DCT) DAN KRIPTOGRAFI *RIVEST CODE* 6 (RC6)

ENDRIKO MARTOFORI
10851002890

Tanggal Sidang: 10 Oktober 2013
Periode Wisuda: November 2013

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Steganografi merupakan sebuah teknik untuk mengamankan komunikasi suatu data. Data berupa pesan, diamankan dengan cara menyisipkan bit-bit pesan ke dalam bit-bit *carrier file.* Salah satu metode untuk menyisipan pesan pada *carrier file* adalah metode *Least Significant Bit* yang menyisipkan bit-bit pesan pada bit-bit terkanan *carrier file*. Laporan penelitian ini membahas tentang rancang bangun aplikasi yang berjalan pada sistem operasi Android untuk mengamankan pesan gambar yang dikirim melalui media pengiriman pesan dengan teknik Steganografi menggunakan Algoritma DCT yang dikolaborasikan dengan Kriptografi menggunakan Algortima RC6 dan disisipkan dengan menggunakan teknik LSB 1 bit. File yang digunakan sebagai media penampung adalah file gambar yang berformat JPEG yang menghasilkan format gambar PNG sebagai keluarannya dan pesan yang akan disisipkan berupa teks. Pengujian yang dilakukan meliputi pengujian pengiriman dan penerimaan pesan gambar yang sudah disisipi pesan . Hasilnya adalah menunjukkan bahwa Algoritma DCT dan RC6 dapat digunakan untuk mengamankan pesan gambar melalui media pengiriman, dimana pesan yang telah disisipi tersebut dapat di dekripsi dengan benar. Dari pengujian dengan menggunakan serangan *exhausiive attack* , diperoleh kesimpulan bahwa data tidak dapat dibuka oleh pihak yang tidak berhak.

Kata kunci: Android, DCT, *Exhausitive Attack*, JPEG, Kriptografi, LSB, PNG, RC6, Steganografi

MESSAGING SAFEGUARDS IN THE PICTURE USING THE DISCRETE COSINE TRANSFORM WITH STEGANOGRAPHY (DCT) AND CRYPTOGRAPHY RIVEST CODE 6 (RC6)

ENDRIKO MARTOFORI
10851002890

Date of Final Exam : July 10th, 2013
Graduation Cremony Period : November 2013

Engineering Departement of Informatic Technology
Faculty of Sciences and Technology
State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

Steganography is a technique to secure a communication data. The Data is messages, which secured by means of inserting the bit-bit messages into bit-bit carrier file. One of method for that is Least Significant Bit method which insert the least right bits of carrier file. This research discusses the architecture of applications that run on the Android operating system to secure the picture message sent through the medium of message delivery by technique of steganography using DCT Algorithms that be collaborationed with the Cryptographic Algorithms using the RC6 and inserted by using the technique of LSB 1 bit. Files used as media is formated JPEG formated file producing as PNG forma picture as deoutput, and message inserted is in the form of text. Testing conducted include testing sending and receiving picture messages that inserted message. The result showed that a DCT Algorithm and RC6 can be used to secure picture messages through the medium of delivery, while the messages inserted can be decrypted correctly. From test by using the exhausiive attack, obtained the conclusion that the data cannot be opened by the part of be not entitled.

Keywords : Android, Cryptography, DCT, Exhausitive Attack, JPEG, LSB, PNG, RC6, Steganography

KATA PENGANTAR

Assalamu ’alaikum wa rahmatullahi wa barakatuh.

Alhamdulillah, puji dan syukur senantiasa diucapkan ke hadirat Allah SWT, atas segala limpahan anugerah dan petunjuk-Nya, Tugas Akhir dengan judul "**PENGAMANAN PENGIRIMAN PESAN DALAM GAMBAR MENGGUNAKAN STEGANOGRAFI DENGAN *DISCRETE COSINE TRANSFORM* (DCT) DAN KRIPTOGRAFI *RIVEST CODE* 6 (RC6)**" ini dapat diselesaikan, sebagai salah satu syarat kelulusan dalam menyelesaikan studi di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.

Banyak sekali pihak yang telah membantu dalam penyelesaian Tugas Akhir ini, baik secara moril maupun materil. Untuk itu, terima kasih dihaturkan kepada:

1. Bapak Prof. DR. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibuk Prof. Dra. HJ. Yennita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibuk Elin Haerani, ST, M.Kom selaku Ketua Jurusan Teknik Informatika.
4. Ibu Luh Kesuma Wardhani, MT selaku Pembimbing I Tugas Akhir yang telah memberikan masukan yang bermanfaat kepada penulis.
5. Pak Rezki Mai Chandra selaku pembimbing II Tugas Akhir ini, yang banyak memotivasi dan membimbing penulis dalam menyelesaikan masalah pada laporan ini.
6. Ibu Lestari Handayani, ST, M.Kom selaku Penguji I Tugas Akhir yang telah memberikan masukan yang bermanfaat kepada penulis.
7. Pak Iwan Iskandar, MT selaku penguji II yang banyak memberikan masukan kepada penulis tentang keamanan data.
8. Pak Benny Sukma Negara, MT Selaku Penasehat Akademis.

9. Pak Ismail Marzuki, ST, yang telah banyak menolong penulis dalam mengerjakan Tugas Akhir ini serta dosen yang pernah menjadi pembimbing II penulis. Semoga sukses kuliah S2 nya di korea ya Pak.
10. Seluruh dosen dan karyawan Fakultas Sains dan Teknologi, khususnya Jurusan Teknik Informatika.
11. Mama dan Papa tercinta yang telah memberikan do'a dan motivasi kepada penulis sehingga Tugas Akhir ini dapat terselesaikan sesuai dengan yang diiinginkan dan tidak lupa dengan adikku tersayang, semoga kita bisa sukses bersama.
12. Lisa Septiany. Terimakasih atas motivasinya yang menyemangati penulis dan telah banyak menolong penulis dalam berbagai hal.
13. Teman-teman Jurusan Teknik Informatika khususnya angkatan 2008, terima kasih atas dukungan, saran, kritik dan diskusinya untuk kesempuranaan penyusunan Tugas Akhir ini.
14. Semua pihak yang tidak bisa disebutkan satu persatu yang telah banyak membantu selama ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, kritik serta saran yang membangun dari rekan-rekan pembaca sangat dibutuhkan agar dapat membuat Tugas Akhir ini lebih baik. Akhir kata penulis berharap agar Tugas Akhir ini bisa memberikan manfaat bagi pembaca dan semua pihak yang berkepentingan. Terima kasih.

Pekanbaru, 10 Oktober 2013

Penulis

DAFTAR ISI

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir ini terjadi beberapa kasus penyadapan dalam bidang komunikasi di dunia ini. Tidak hanya telepon tetapi juga penyadapan dalam bentuk (Short Message Service) SMS sehingga meresahkan banyak kalangan masyarakat yang tidak ingin diketahui pesannya. Pada umumnya penyadapan ini dilakukan oleh pemerintah, yang bertujuan untuk mengatahui pesan rahasia seorang yang dianggap tersangka dalam tindak kejahatan baik korupsi, teroris, dan lain - lain. Tetapi penyadapan ini berpotensi merugikan hak asasi manusia, karena pesan SMS itu bersifat pribadi dan sensitive. Seperti yang dilakukan oleh (Komisi Pemberantasan Korupsi) KPK dalam menyadap beberapa tersangka korupsi seperti, Al Amin Nasution dalam kasus korupsi yang dikenal dengan "skandal gadis berbaju putih" dan yang masih segar dalam ingatan kita, dimana Antasari Azhar dalam kedudukan sebagai ketua KPK memerintahkan penyadapan terhadap Nasrudin Zulkarnaen, Direktur PT Putra Rajawali Banjaran, dimana ternyata Nasrudin Zulkarnaen tidak terlibat dalam kasus korupsi. Kasus – kasus ini sebenarnya sudah cukup menunjukkan betapa mudahnya hak privasi dari seorang warga negara diganggu oleh negara meski dilakukan dalam bungkus upaya penegakkan hukum.

Sebagai warga negara Indonesia kita mempunyai hak dilindungi dalam privasi kita seperti yang ditegaskan dalam Pasal 28 G ayat (1) UUD 1945 telah menyebutkan "Setiap orang berhak atas perlindungan diri pribadi, keluarga, kehormatan, martabat, dan harta benda yang di bawah kekuasaanya, serta berhak atas rasa aman dan perlindungan dari ancaman ketakutan untuk berbuat atau tidak berbuat sesuatu yang merupakan hak asasi". Tetapi, hingga saat ini tidak ada formulasi hukum yang tepat untuk mengimplementasikan ketentuan Pasal 28 G ayat (1) UUD 1945 ini.

Dengan lemahnya keamanan privasi yang kita dapat dari pemerintah, maka informasi atau pesan yang dikirim, agar tidak diketahui oleh pihak lain isi dari informasi atau pesan tersebut, selain pihak pengirim dan penerima dibutuhkan sebuah teknik untuk menjaga kerahasiaan tersebut. Teknik tersebut adalah kriptografi, yaitu isi informasi yang asli (*plainteks*) diubah menjadi informasi acak (*cipherteks*) terlebih dahulu, yang sama sekali tidak memiliki makna. Jika ada pihak ketiga menyadap dan mengambil pesan yang dikirim maka pesan tersebut tidak akan dapat dibaca,karena pesan tersebut sudah teracak yang pastinya akan sulit dimengerti oleh penyadap.

Namun teknik kriptografi memiliki kelemahan, yaitu pesan acak yang dikirim justru dapat menimbulkan kecurigaan oleh pihak luar, sehingga pesan tersebut dapat dirusak dengan tujuan agar pihak penerima yang asli tidak berhasil mendapatkan pesan tersebut secara utuh. Untuk mengatasi hal ini, dapat digunakan teknik lain, yaitu steganografi. Dengan metode ini, isi informasi yang ingin dikirimkan disembunyikan/disisipkan ke dalam suatu bentuk media yang umum (*cover-object*) dan dapat kita temui sehari-hari. Hasil dari penyisipan ini (*stego-object*) akan dapat dikirimkan dengan aman, karena apabila terdapat pihak luar yang menyadap, informasi yang didapat berupa media yang umum sehingga kemungkinan untuk timbulnya kecurigaan sangatlah kecil.

Steganografi merupakan salah satu bagian dari kriptografi, yaitu ilmu dan seni dalam menyembunyikan pesan rahasia sedemikian sehingga manusia tidak dapat menyadari keberadaan pesan tersebut. Pada masa kini, steganografi lebih banyak dilakukan pada data *digital*, dengan menggunakan bentuk media *digital* seperti teks, gambar, audio, atau video (Munir, 2006).

Berbagai penelitian terhadap steganografi ini sebenarnya telah dilakukan dan tetap dikembangkan oleh para peneliti menggunakan beragam metode steganografi. Teknik penyisipan pesan yang digunakan dalam pembuatan laporan Tugas Akhir ini adalah *DCT (Discrete Cosine Transform)*, yang bekerja pada domain frekuensi. DCT adalah proses untuk mengubah domain spasial gambar menjadi domain frekuensi, dan digunakan pada gambar berformat JPEG. Penggunaan teknik DCT, pada laporan ini disebabkan karena penurunan kualitas

gambar yang dihasilkan tidak terlalu signifikan, dan pesan di dalamnya akan tidak hilang apabila dilakukan perubahan terhadap gambar tersebut.

Beberapa contoh jurnal dan penelitian yang berhubungan dengan DCT ini diantaranya adalah: Andrew B. Watson dan NASA Ames Research Center (1994) Image Compression Using the Discrete Cosine Transform dan Paul Gunawan Hariyanto (2008) Studi dan Implementasi Steganografi pada Video *Digital* di *Mobile Phone* dengan *DCT Modification*. Pada Tugas Akhir ini perbedaan yang ada pada jurnal dan penelitian sebelumnya yaitu pada metode kriptografi yang digunakan, metode kriptografi yang digunakan oleh Andrew B. Watson dan NASA Ames Reseach Center adalah Metode *entropy coding* yang dibangun menggunakan bahasa pemprogaraman java berbasis desktop, sedangkan pada Paul Gunawan Hariyanto mengguanakan Algoritma MD5 dan LCG yang berfungsi sebagai bilangan acak dan aplikasinya dibangun menggunakan bahasa java bebasis *mobile* (J2ME). Aplikasi yang akan dibangun pada Tugas Akhir ini menggunakan kriptografi RC6.

Aplikasi yang akan dibangun pada Tugas Akhir ini merupakan aplikasi pada perangkat *mobile phone*. Format gambar yang digunakan sebagai media penyimpanan adalah format JPEG, yang juga merupakan format gambar yang paling banyak didukung *mobile phone* saat ini. Format JPEG dapat dikirimkan melalui layanan MMS (*Multimedia Messaging Service*) pada *mobile phone*.

1.2 Rumusan Masalah

Berdasarkan penjelasan yang telah dijelaskan pada bagian latar belakang di atas, maka didapat sebuah rumusan masalah yang akan dijelaskan lebih lanjut pada tugas akhir ini, yaitu bagaimana membangun sebuah aplikasi pengamanan pengiriman pesan dalam gambar menggunakan teknik steganografi dengan metode DCT dan kriptografi RC6

1.3 Batasan Masalah

Agar tidak terjadi kesalahan persepsi dalam Tugas Akhir ini, maka dijelaskan batasan masalah pada laporan ini yaitu :

1. Media penampung yang digunakan adalah file citra yang berformat JPEG dengan keluaran file yang berformat PNG.

2. Bit-bit *carrier file* yang akan dimodifikasi adalah hanya pada 1 bit LSB saja.
3. Pesan yang disisipi berupa teks file.

1.4 Tujuan

Adapun tujuan dari Tugas Akhir ini adalah membangun sebuah aplikasi yang aman untuk pengiriman pesan dalam gambar dengan menggunakan metode DCT dan RC6.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, meliputi *Mobile Phone*, Android, Steganografi, Ktiptografi, dan format gambar.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini merupakan pembahasan tentang analisis perangkat lunak, meliputi analisis, analisis masalah, analisis metode, analisis kebutuhan sistem, serta perancangan. Perancangan sistem yang terdiri dari perancangan diagram alir (*flowchart*).

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi yang dilakukan terhadap steganografi pada gambar di *mobile phone* berbasis android dengan DCT *Modification* dan kriptografi RC6.

BAB VI PENUTUP

Bab ini berisi kesimpulan yang dihasikan dari pembahasan tentang pengiriman pesan gambar menggunakan steganografi dengan metode DCT dan kriptogarafi RC6 dan saran sebagai hasil akhir dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1. Teori Keamanan Data

Data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter, dapat berupa alfabet, angka, maupun simbol khusus seperti *, $ dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data. Informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan.

Aman juga sering diartikan dengan istilah free from danger yang artinya bebas dari ancaman bahaya. Keamanan informasi adalah cabang studi dari teknologi informasi yang mengkhususkan diri untuk mempelajari metode dan teknik untuk melindungi informasi dan sistem informasi dari akses, penggunaan, penyebaran, perusakan, perubahan, dan penghancuran tanpa otorisasi yang sah.

Ada empat aspek utama dalam keamanan data dan informasi, yaitu :

1. Privacy/Confidentiality yaitu usaha menjaga data informasi dari orang yang tidak berhak mengakses.
2. Integrity yaitu usaha untuk menjaga data atau informasi tidak diubah oleh yang tidak berhak.
3. Authentication yaitu usaha atau metoda untuk mengetahui keaslian dari informasi, misalnya apakah informasi yang dikirim dibuka oleh orang yang benar (asli) atau layanan dari server yang diberikan benar berasal dari server yang dimaksud.
4. Availability berhubungan dengan ketersediaan sistem dan data (informasi) ketika dibutuhkan.

Keempat aspek ini menjadi dasar untuk m elakukan pengamanan terhadap data dan informasi.

2.1.1. Pengenalan Steganografi

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui. Steganografi berasal dari bahasa Yunani, yaitu "steganos" yang artinya "tulisan tersembunyi (covered writing)". steganografi dapat dipandang sebagai kelanjutan kriptogarafi dan dalam prakteknya pesan rahasia dienkripsi terlebih dahulu, kemudian cipherteks disembunyikan di dalam media lain sehingga pihak ketiga tidak menyadari keberadaannya. Pesan rahasia yang disembunyikan dapat diekstraksi kembali persis sama aslinya.

Steganografi membutuhkan dua property yaitu media penampung dan pesan rahasia. Media penampung yang umum digunakan adalah gambar, suara, video, atau teks. Pesan yang disembunyikan dapat berupa sebuah artikel, gambar, daftar barang, kode program, atau pesan lain.

Keuntungan steganografi dari kriptografi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung yang membawa pesan tidak menimbulkan kecurigaan bagi pihak ketiga. Ini berbeda dengan kriptografi yang mana cipherteks menimbulkan kecurigaan bahwa pesan tersubut merupakan pesan rahasia.

A. Konsep dan Terminologi

Terdapat beberapa istilah yang berkaitan dengan steganografi :

1. *Hiddentext* atau *embedded message* yaitu pesan yang disembunyikan.
2. *Covertext atau cover-object* yaitu pesan yang digunakan digunakan untuk menyembunyikan *embedded message*.
3. *Stegotext* atau *stego-object* yaitu pesan yang sudah berisi *embedded message*.

Di dalam steganografi digital, baik *hiddentext* maupun *covertext* dapat berupa teks, citra, audio maupun video. Jadi kita dapat menyembunyikan pesan berupa kode program di dalam sebuah citra atau di dalam video, atau kita juga

dapat menyembunyikan gambar rahasia di dalam citra laian atau di dalam sebuah berkas music *mp3*.

Penyisipan pesan kedalam media *covertext* dinamakan *encoding*, sedangkan ekstrasi pesan dari *stegotext* dinamakan *decoding*. Kedua proses ini mungkin memerlukan kunci rahasia (*stegokey*) agar hanya pihak yang berhak saja yang dapat melakukan penyisipan pesan dan ekstrasi pesan.

Penyembunyian pesan rahasia ke dalam media penampung pasti mengubah kualitas media tersebut. Criteria yang harus diperhatikan dalam penyembunyian pesan adalah :

1. *Imperceptibility* : Keberadaan pesan tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext* nya. Jika, *covertext* berupa audio maka indera telinga tidak dapat mendeteksi perubahan pada audio *stegotext* nya.
2. *Fidelity* : Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan tersebut tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext* nya. Jika *covertext* berupa audio maka audio *stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan tersebut.
3. *Recovery* : Pesan yang disembunyikan harus dapat diungkapkan kembali. Karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diamabil kembali untuk digunakan lebih lanjut.

B. Metode-Metode Steganografi

Ada beberapa metode yang dapat digunakan sebagai teknik penyembunyian suatu informasi digital ke dalam informasi digital lainnya *(steganography)*, diantaranya adalah:

1. *Least Significant Bit Insertion* (LSB)

   LSB merupakan sebuah metode yang lazim digunakan oleh para peneliti pada sebuah steganografi. Hal ini disebabkan karena LSB merupakan

sebuah metode steganografi yang paling sederhana, cepat, dan mempunyai kapasitas penyisipan suatu informasi digital yang cukup besar. LSB menyisipkan sebuah informasi rahasia pada bit rendah atau bit yang paling kanan dari sebuah data pixel yang menyusun sebuah informasi digital yang menjadi media penampung suatu informasi rahasia.

2. *Masking and Filtering*

   Metode ini biasanya dibatasi pada image 24 bit warna dan *image grayscale*. Beberapa literatur menyatakan bahwa metode ini mirip dengan *watermark*, dimana suatu image diberi tanda *(marking)* untuk menyembunyikan pesan rahasia. Hal ini dapat dilakukan dengan memodifikasi *luminance image* dibeberapa bagiannya. Metode ini memiliki ketahanan *(robustness)* terhadap kompresi, dan *cropping*. Namun, memiliki batasan kapasitas tertentu pada informasi yang akan disembunyikan.

3. *Algorithm and Transformation*

   Metode ini merupakan metode steganografi yang jauh lebih kompleks dari metode-metode sebelumnya, artinya tingkat kesulitan dalam penerapan metode ini lebih tinggi. Untuk menyembunyikan sebuah informasi digital pada media penampungnya dilakukan dengan memanfaatkan *Discerete Cosine Transformation* (DCT) dan *Wavelet Compression*. DCT digunakan pada file-file terkompresi, seperti JPEG. Metode ini terjadi di domain frekuensi dari sebuah file digital, bukan pada domain spasial.

4. *Spread Spectrum Methode*

   Teknik metode ini dalam menyembuyikan suatu informai digital adalah dengan mengkodekan informasi rahasia dan disebarkan ke setiap spektrum frekuensi yang memungkinkan. Namun, metode ini masih mudah diserang, yaitu dengan cara penghancuran atau pengrusakan dari kompresi dan proses image (gambar).

2.2. Kompresi Citra

Kompresi adalah pengubahan data kedalam bentuk yang memerlukan bit yang lebih sedikit, biasanya dilakukan agar data dapat disimpan atau dikirimkan

dengan lebih efisien. Jika kebalikan dari proses ini, yaitu dekompresi, menghasilkan data yang sama persis dengan data aslinya, maka kompresi tersebut disebut *lossless compression*. Sebaliknya, dekompresi tersebut menghilangkan sebagian data, maka disebut *loosy compression*. *Loosy compression* biasanya diterapkan dalam kompresi data berupa gambar. Walaupun tidak dapat menghasilkan data yang sama persis dengan aslinya, namun dianggap lebih efisien.

2.2.1. Teknik Kompresi Citra

A. Lossy Compression

Dengan teknik ini ukuran file citra menjadi lebih kecil, dengan menghilangkan beberapa informasi dalam citra asli. Teknik ini mengubah detail dan warna pada file citra menjadi lebih sederhana tanpa terlihat perbedaan yang mencolok dalam pandangan manusia, sehingga ukurannya menjadi lebih kecil. Biasanya digunakan pada citra foto atau image lain yang tidak terlalu memerlukan detail citra, dimana kehilangan bit rate foto tidak berpengaruh pada citra. Adapun macam – macam metode lossy yaitu,

1. Color reduction yaitu untuk warna-warna tertentu yang mayoritas dimana informasi warna disimpan dalam color palette.
2. Chroma subsampling yaitu teknik yang memanfaatkan fakta bahwa mata manusia merasa brightness (luminance) lebih berpengaruh daripada warna (chrominance) itu sendiri, maka dilakukan pengurangan resolusi warna dengan disampling ulang. Biasanya digunakan pada sinyal YUV. Chorma Subsampling terdiri dari 3 komponen yaitu, Y (luminance) : U (CBlue) : V (CRed)
3. Transform coding yaitu menggunakan Fourier Transform seperti DCT.
4. Fractal Compression: adalah suatu metode lossy untuk mengkompresi citra dengan menggunakan kurva fractal. Sangat cocok untuk citra natural seperti pepohonan, pakis, pegunungan, dan awan. Fractal Compression bersandar pada fakta bahwa dalam sebuah image, terdapat bagian-bagian image yang menyerupai bagian bagian image yang lain. Proses kompresi

Fractal lebih lambat daripada JPEG sedangkan proses dekompresinya sama.

B. Loseless Compression

Teknik kompresi citra dimana tidak ada satupun informasi citra yang dihilangkan. Biasanya digunakan pada citra medis. Adapun macam - macam metode loseless yaitu, Run Length Encoding, Entropy Encoding (Huffman, Aritmatik), dan Adaptive Dictionary Based (LZW)

2.2.2. Hal Penting Dalam Kompresi Citra

A. **Scalability/Progressive Coding/Embedded Bitstream**

Adalah kualitas dari hasil proses pengkompresian citra karena manipulasi bitstream tanpa adanya dekompresi atau rekompresi. Biasanya dikenal pada loseless codec. Contohnya pada saat preview image sementara image tersebut didownload. Semakin baik scalability, makin bagus preview image. Tipe scalability yaitu, Quality progressive, dimana image dikompres secara perlahan-lahan dengan penurunan kualitasnya, Resolution progressive, dimana image dikompresi dengan mengenkode resolusi image yang lebih rendah terlebih dahulu baru kemudian ke resolusi yang lebih tinggi dan Component progressive, dimana image dikompresi berdasarkan komponennya, pertama mengenkode komponen gray baru kemudian komponen warnanya.

B. Region of Interest Coding

Yang mana daerah-daerah tertentu dienkode dengan kualitas yang lebih tinggi daripada yang lain.

C. **Meta Information**

Image yang dikompres juga dapat memiliki meta information seperti statistik warna, tekstur, small preview image, dan author atau copyright information

2.2.3. Algortima Kompresi JPEG

JPEG singkatan dari Joint Photographic Experts Group, sebuah komite standar yang memiliki asal-usul dalam Standar Internasional Organization (ISO). JPEG menyediakan metode kompresi yang mampu mengompresi data gambar

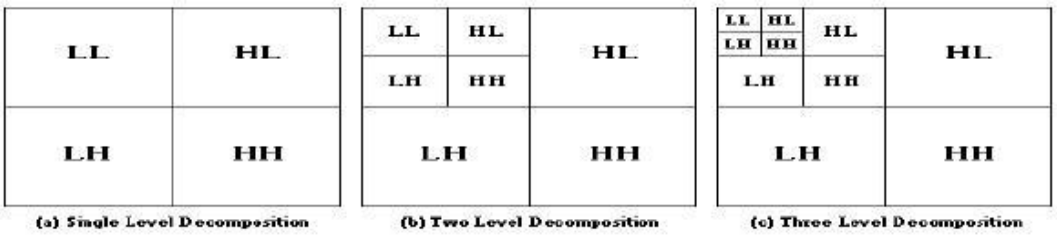
piksel dengan kedalaman 6 sampai 24 bit dengan kecepatan yang wajar dan efficiency. JPEG dapat disesuaikan untuk menghasilkan yang sangat kecil gambar terkompresi yang berkualitas relatif miskin dalam penampilan tetapi masih cocok untuk banyak aplikasi. Sebaliknya, JPEG mampu menghasilkan kompresi gambar yang sangat tinggi yang masih jauh lebih kecil dari data yang tidak terkompresi.

JPEG adalah metode kompressi lossy yang utama. JPEG dirancang khusus untuk membuang informasi bahwa mata manusia tidak dapat dengan mudah melihat. Sedikit perubahan dalam warna tidak dianggap baik oleh mata manusia, daripada perubahan-perubahan kecil dalam intensitas (terang dan gelap). Oleh karena itu encoding lossy JPEG cenderung menjadi lebih hemat dengan bagian skala abu-abu gambar dan menjadi lebih dangkal dengan warna. DCT memisahkan gambar menjadi bagian-bagian frekuensi yang berbeda dimana frekuensi kurang penting dibuang melalui kuantisasi dan frekuensi penting digunakan untuk mengambil gambar selama dekompresi. Dibandingkan dengan transform dependent yang lain, DCT memiliki banyak keuntungan:

a. Telah diimplementasikan dalam sirkuit terpadu,
b. Memiliki kemampuan untuk mempaket semua informasi dalam koefisien yang paling sedikit,
c. Meminimalkan penampilan blok seperti yang disebut memblokir artefak yang terjadi ketika batas antara sub-gambar menjadi terlihat.

A. Algoritma Umum Untuk Kompresi Image

1. Menentukan bitrate dan toleransi distorsi image dari inputan user.
2. Pembagian data image ke dalam bagian-bagian tertentu sesuai dengan tingkat kepentingan yang ada (classifying). Menggunakan salah satu teknik: DWT (Discreate Wavelet Transform) yang akan mencari frekuensi nilai pixel masing-masing, menggabungkannya menjadi satu dan mengelompokkannya sebagai berikut:



Gambar 2.1 Pembagian Data Image Dengan Teknik DWT

Dimana :

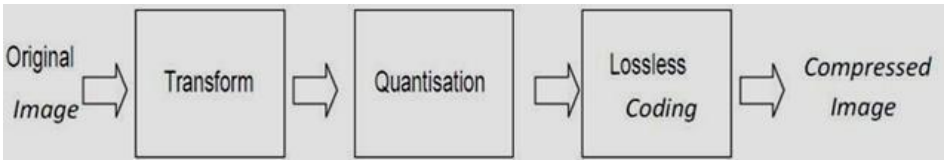
LL : Low Low Frequency (most importance)

HL : High Low Frequency (lesser importance)

LH : Low High Frequency (more lesser importance)

HH : High High Frequency (most less importance)

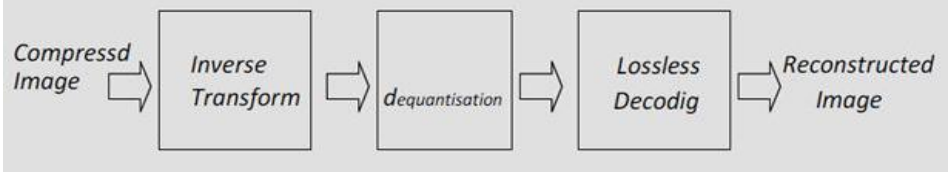
3. Pembagian bit-bit di dalam masing-masing bagian yang ada (bit allocation).
4. Lakukan kuantisasi (quantization).
5. Lakukan pengenkodingan untuk masing-masing bagian yang sudah dikuantisasi tadi dengan menggunakan teknik entropy coding (Huffman dan aritmatik) dan menuliskannya ke dalam file hasil.



Gambar 2.2 Model Kompresi Citra

B. Algoritma Umum Dekompresi Image

1. Baca data hasil kompresi menggunakan entropy dekoder.
2. Dekuantisasi data.
3. Rebuild image.



Gambar 2.3 Model Dekompresi Citra

2.2.4. Proses Kompresi JPEG Menggunakan DCT

A. DCT (Discrete Cosine Transform)

DCT (Discrete Cosine Transform) merupakan salah satu teknik transformasi yang mengubah suatu sinyal menjadi unsur komponen frekuensi. DCT pertama kali oleh Ahmed, Natarajan dan Rao pada tahun 1974 dalam makalahnya yang berjudul "*On Image processing and a discrete cosin transform*" (Watson, 1994).

DCT pada kompresi JPEG menerima masukan berupa matriks gambar berukuran 8x8, yang kemudian mengubahnya menjadi matriks frekuensi dengan ukuran yang sama. Sedangkan proses IDCT yang merupakan kebalikan dari DCT, akan mengembalikan koefisien pada matriks frekuensi menjadi matriks gambar. Persamaan DCT dapat dilihat pada rumus berikut :

$$F(u,v) = \frac{1}{4}.C(u).C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y).\cos\frac{(2x+1)u\pi}{16}.\cos\frac{(2y+1)v\pi}{16}\right]$$

(2.1)

Dengan u = 0, 1, 2, …, N-1, dan v = 0, 1, 2, …, M-1, sedangkan

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{untuk } k = 0 \\ 1 & \text{untuk } k \neq 0 \end{cases}$$

(2.2)

Dimana :

F (u,v) = data pada domain frekuensi
f (x,y) = data pada domain ruang
u, v = koordinat pixel untuk blok transformasi
x, y = koordinat pixel untuk citra sebelum transformasi
F (u) = nilai dari koefisien domain transformasi pada koordinat u
F (v) = nilai dari koefisien domain transformasi pada koordinat v
n = jumlah baris dalam blok yang akan ditransformasikan
m = jumlah baris dalam blok yang akan ditransformasikan

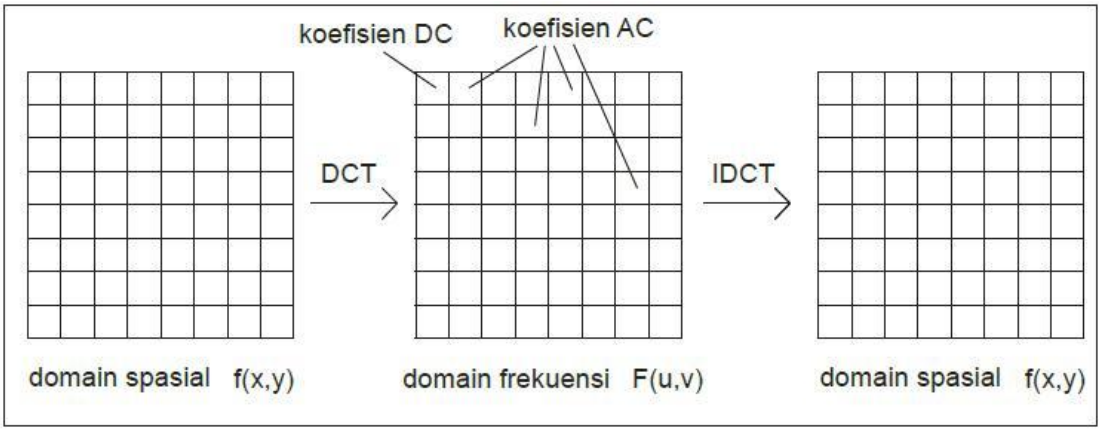
Persamaan IDCT adalah sebagai berikut :

$$f(x,y) = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u).C(v).F(u,v).\cos\frac{(2x+1)u\pi}{16}.\cos\frac{(2y+1)v\pi}{16}\right]$$

(2.3)

dengan x = 0, 1, 2, …, N-1, dan y = 0, 1, 2, …, M-1

terdapat dua jenis koefisien pada matriks frekuensi, yaitu koefisien DC dan AC. Koefisien DC merupakan nilai pada frekuensi 0. Jumlah koefisien ini hanya satu, yang terletak pada sudut kiri atas matriks frekuensi. Sedangkan 63 koefisien lainnya merupakan koefisien AC yang frekuensinya lebih besar dari 0, semakin ke kanan maka menunjuk pada frekuensi horizontal yang semakin tinggi, semakin ke bawah maka akan menunjuk pada frekuensi vertical yang semakin tinggi. Gambar dibawah ini menunjukan letak koefisien DC dan AC, serta alur proses DCT dan IDCT pada perubahan domain gambar :

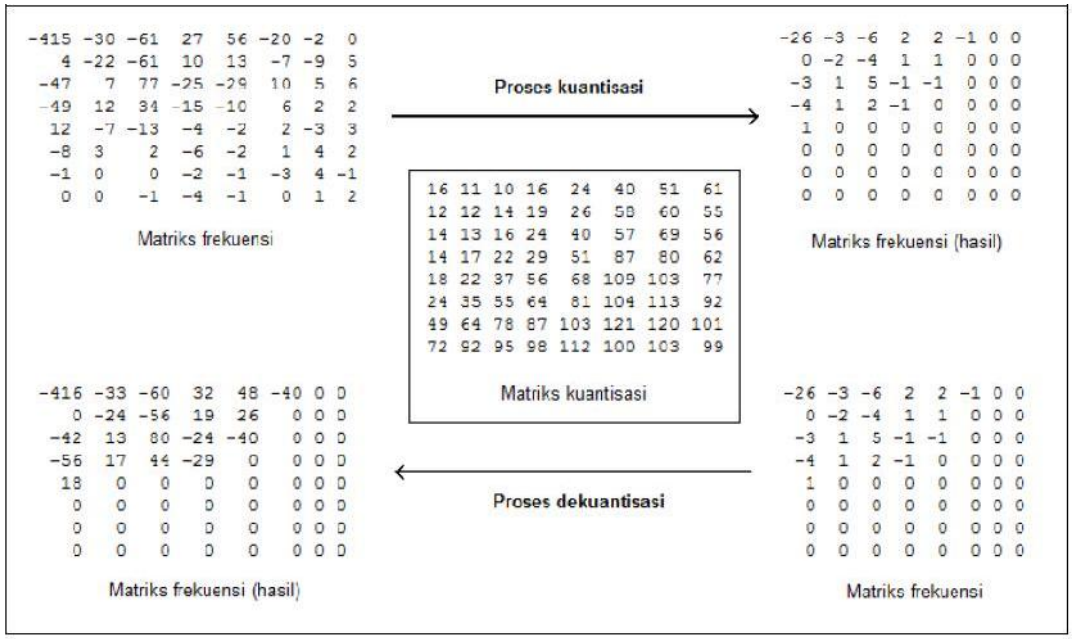


Gambar 2.4 Alur DCT dan IDCT

Pada kebanyakan gambar, nilai koefisien pada frekuensi tinggi benilai kecil, sehingga pengaruhnya pada gambar juga kecil. Dengan membuang nilai pada frekuensi tinggi, dan menyimpan nilai pada frekuensi rendah proses kompresi dapat dilakukan. Proses pemotongan nilai ini dinamakan proses kuantisasi, yaitu membagi matriks frekuensi dengan suatu nilai yang disebut sebagai matriks kuantisasi. Hasil yang diharapkan adalah nilai frekuensi tinggi pada matriks akan menjadi 0 .

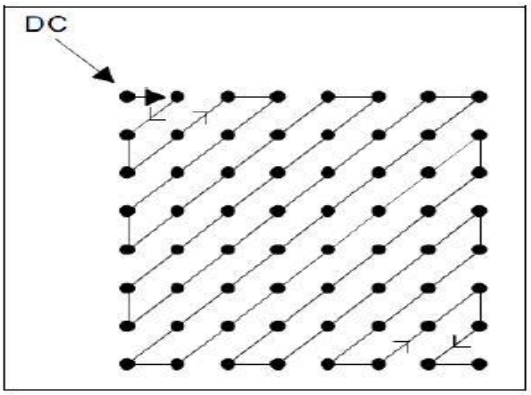
Untuk menampilkan gambar kembali, dilakukan proses dekuantisasi sebagai kebalikan dari proses ini, yaitu mengalikan nilai matriks frekuensi dengan

matriks kuantisasi. Pemilahan nilai pada matriks kuantisasi dibebaskan pada encoder, dimana semakin besar nilainya maka kompresi akan semakin tinggi, sekaligus akan menurunkan kualitas gambar. Salah satu matriks kuantisasi yang banyak digunakan [CCI93], terdapat pada contoh proses kuantisasi dibawah ini :



Gambar 2.5 Contoh Proses Kuantisasi dan Dekuantisasi

Matriks frekuensi yang telah di kuantisasi biasanya banyak memiliki nilai 0 di bagian bawah kanan. Proses kompresi dilanjutkan dengan melukukan entropy coding untuk menyimpan matriks dalam bentuk zig-zag. Dengan cara ini nilai 0 akan terkumpul berurut sehingga penyimpanan nilai matriks ini dapat di singkat, seperti gambar dibawah ini :



Gambar 2.6 Urutan Zig-Zag Pada *Entropy Coding*

B. Alur Algoritma Menggunakan DCT Pada Kompresi JPEG

1. Gambar asli dibagi menjadi blok-blok 8 x 8.
2. Nilai-nilai pixel dari berbagai gambar hitam dan putih adalah antara 0-255 tapi DCT dirancang untuk bekerja pada nilai-nilai piksel mulai dari -128 sampai 127. Oleh karena itu setiap blok dimodifikasi untuk bekerja dalam kisaran.
3. Persamaan digunakan untuk menghitung DCT matriks.
4. DCT diterapkan untuk setiap blok dengan mengalikan blok dimodifikasi dengan DCT matriks di sebelah kiri dan transpos dari matriks DCT di sebelah kanannya.
5. Setiap blok kemudian dikompresi melalui kuantisasi. Matriks terkuantisasi kemudian dikodekan entropi.
6. Gambar yang terkompresi direkonstruksi melalui proses terbalik. Invers DCT digunakan untuk dekompresi.

2.3. Sejarah Kriptografi

Kriptografi sudah ada sebelum adanya computer. Julius cesar, yang khawatir pesan yang dikirim untuk para jendralnya jatuh ke tangan musuh, maka ia menggunakan metode enkripsi sederhana dengan menggeser huruf pada abjad dengan nilai tertentu.

Sepanjang sejarah pembentukan kode dan pemecahannya selalu mendapat perhatian khusus dalam operasi militer. Penggunaan computer untuk pertama kalinya dalam kriptografi merupakan usaha untuk memecahkan kode enigma Nazi sewaktu perang dunia II.

2.3.1. Enkripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai sebuah kode atau *chipper*. Sebuah chipper menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti (*unnitelligible*).

Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak. Infromasi ini dapat berupa nomor kartu kredit, catatan penting dalam computer, maupun *password* untuk mengakses sesuatu.

A. Kategori Enkripsi

Ada tiga kategori dalam enkripsi, yaitu :

1. Kunci enkripsi rahasia.

   Dalam hal ini, terdapat sebuah kunci yang digunakan untuk mengenkripsi dan juga sekaligus mendeskripsikan informasi

2. Kunci enkripsi publik.

   Dalam hal ini menggunakan dua kunci, satu untuk proses enkripsi dan satu lagi untuk proses dekripsi.

3. Fungsi one-way

   atau fungsi satu arah adalah fungsi di mana informasi di enkripsi untuk menciptakan “signature” dari informasi asli yang bisa digunakan untuk keperluan autentikasi.

2.3.2. Pengenalan Kriptografi RC6

Algoritma RC6 merupakan salah satu kandidat *Advanced Encryption Standard* (AES) yang diajukan oleh RSA Security Laboratories kepada NIST. Dirancang oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. RC6 adalah algoritma yang menggunakan ukuran blok hingga 128 bit, dengan ukuran kunci yang diguankan bervariasi antara 128, 192 dan 256 bit.

Algoritma RC6 dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b. Parameter w merupakan ukuran kata dalam satuan bit, parameter merupakan bilangan bukan negatif yang menunjukan banyaknya iterasi selama proses enkripsi dan parameter b menunjukan ukuran kunci enkripsi dalam byte. Setelah algoritma ini masuk dalam kandidat AES, maka ditetapkan bahwa nilai w = 32, r=20 dan b bervariasi antara 16, 24 dan 32 byte.

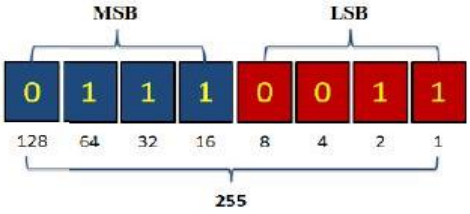
RC6-w/r/b memecah blok 128 bit menajdi 4 buah blok 32-bit, dan mengikuti aturan enam operasi dasar sebagai berikut :

1. a + b operasi penjumlahan bilangan integer
2. a - b operasi pengurangan bilangan integer
3. a ⊕ b operasi exclusive-OR (XOR)
4. a x b operasi perkalian bilangan integer
5. a<<<b a dirotasikan ke kiri sebanyak variabel kedua (b)
6. a>>>b a dirotasikan ke kanan sebanyak variabel kedua (b)

2.4. Konsep Kerja LSB *(Least Significant Bit)*

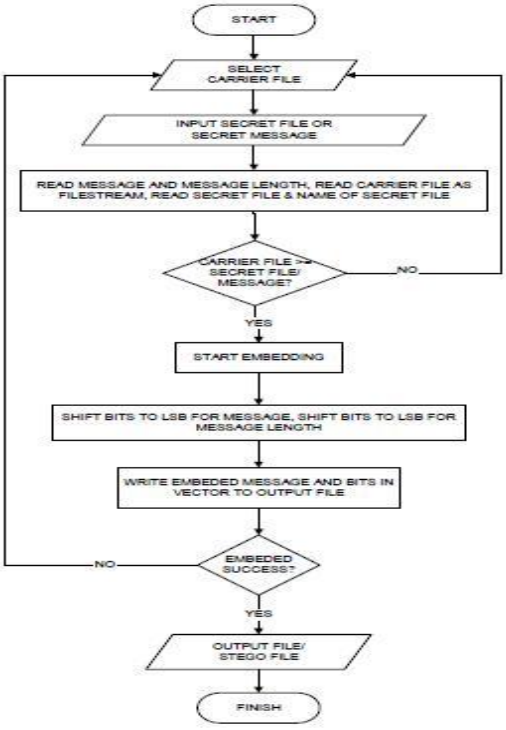
Bilangan biner merupakan dasar dari terciptanya komputer, karena sebenarnya komputer bekerja berdasarkan dua bilangan saja, yaitu 0 dan 1. Kedua bilangan ini sering disebut dengan istilah *bit.* Kemudian bit-bit ini akan terus berangkai dan bersusun membentuk suatu struktur biner yang menjadi sebuah rangkaian informasi. Bentuk yang paling umum digambarkan untuk serangkaian bit berjumlah 8-bit atau sering disebut dengan istilah 1 byte *(Alfebra dkk, 2008).*

Pada sebuah rangkaian informasi terdapat penggolongan-penggolongan bit berdasarkan urutan dan pengaruhnya dalam byte. Secara garis besar, dalam rangkaian informasi terdapat 2 golongan bit, yaitu *Most Significant Bit* (MSB) dan *Least Significan Bit* (LSB), seperti yang ditunjukkan pada gambar 2.2. *Most Significant Bit* merupakan reprsentasi 4-bit yang memiliki pengaruh besar pada sebuah rangakain informasi, artinya adalah akan terjadi perubahan yang drastis apabila bit-bit ini dimodifikasi. Sementara *Least Significant Bit* merupakan representasi 4-bit yang paling sedikit memiliki pengaruh apabila bit-bit tersebut dimodifikasi dan tidak akan terjadi perubahan yang drastis, sehingga kemungkinan terjadinya kecurigaan manusia terhadap bit-bit LSB yang dimodifikasi sangat kecil. Dengan demikian, semakin kekanan, bit-bit tersebut semakin kecil pengaruhnya terhadap keutuhan data yang dikandung. Oleh sebab itu, 4-bit terakhir tersebut yang dimodifikasi dan dijadikan tempat pelekatan sebuah informasi digital steganografi.



Gambar 2.7 Representasi Biner *(Sumber: Agus dkk, 2010)*

Teknik LSB dilakukan dengan memodifikasi bit-bit yang tergolong bit-bit LSB pada tiap byte dalam sebuah *file* yang digunakan sebagai *carrier file*, atau dengan kalimat yang lain dengan cara mengganti bit-bit LSB dengan bit-bit informasi yang ingin dilekatkan. Proses penggantian bit ini disebut dengan proses *encoding/ embedding.* Setelah semua bit informasi tersebut menggantikan bit LSB *carrier file* tersebut, maka informasi telah berhasil dilekatkan pada *carrier file* dan *output*-nya disebut dengan *Stegofile.* Apabila suatu informasi yang dilekatkan tersebut ingin dibuka *(ekstract)* kembali, maka bit-bit LSB yang ada pada *stegofile* akan diambil satu per satu dan dikembalikan lagi atau disatukan kembali sehingga menjadi sebuah informasi atau disebut dengan *decoding/ retrieving (Alfebra dkk, 2008).* Standar *flowchart* dari metode LSB yang digunakan untuk menyisipkan informasi dan mengekstraknya dapat dilihat pada gambar 2.8.



Gambar 2.8 Flowchart LSB *(Sumber: Alfebra dkk, 2008)*

Berdasarkan Gambar 2.8 dapat dijelaskan alur steganografi dalam menyembunyikan sebuah data digital dengan metode LSB:

1. Pertama sekali yang perlu dilakukan adalah menginputkan sebuah data digital yang berfungsi sebagai media penampung penyembunyian data digital lain *(carrier file).*
2. Kemudian pilih file atau teks yang akan disembunyikan dalam *carrier file* dengan cara mengonversikannya ke bentuk biner.
3. Lakukan pengukuran ukuran file atau teks yang akan disembunyikan dimana file dan teks harus >= dari file atau teks yang akan disembunyikan.
4. Selanjutnya, proses *embedding* dilakukan.
5. Proses *embedding* dilakukan dengan menukar bit-bit terakhir (LSB) pada *carrier file* diganti dengan bit-bit pada teks atau file yang akan disembunyikan.
6. Konversikan *carrier file* yang telah dimodifikasi dengan bit-bit file atau teks ke bentuk vektor. Jika sukses, maka akan dihasilkan file steganografi *(stegofile).*
7. Selesai.

Berikut ini adalah contoh proses steganografi. Pada contoh dimisalkan bilangan biner pada kotak berikut adalah *carrier file* untuk pesan yang akan disembunyikan. Anggap pesan yang akan disembunyikan adalah teks.

01001101 00101110 10101110 10001010 10101111 10100010 00101011 10101011

Biner-biner diatas digunakan untuk menyimpan karakter "**H**" yang memiliki nilai biner (**01001000**), maka pixel-pixel wadah tersebut akan dirubah menjadi :

01001100 00101111 10101110 10001010 10101111 10100010 00101010 10101010

2.5. Teknologi Telekomunikasi Telepon Selular

Ponsel merupakan gabungan dari Teknologi Radio yang dikawinkan dengan Teknologi Komunikasi Telepon. Telepon pertama kali ditemukan dan diciptakan oleh Alexander Graham Bell pada tahun 1876. sedangkan komunikasi tanpa kabel (wireless) ditemukan oleh Nikolai Tesla pada tahun 1880 dan diperkenalkan oleh Guglielmo Marconi.

Perkembangan teknologi *wireless* yang sedang berkembang pesat saat ini yaitu teknologi telepon tanpa kabel (*wireless*) diantaranya AMPS (*Advance Mobile Phone System*), GSM (*Global System for Mobile system*) dan CDMA (*Code Division Multiple Access*).

A. AMPS (*Advance Mobile Phone System*)

AMPS merupakan generasi pertama pada teknologi selular. System ini di alokasikan pada Band 800 Mhz. jaringan ini mengguakan sirkuit terintergrasi yang sangat besar yang terdiri dari *Computer Dedicated* dan *System Switch*.

AMPS menggunakan range frekuensi antara 824 Mhz – 894 Mhz yang diperuntukan pada ponsel analog. AMPS hanya di operasikan pada band 800 Mhz dan tidak menawarkan fitur lain yang umum digunakan pada layanan seluler seperti e-mail dan *browsing* di web. Kualitas suara yang kurang bagus serta beberapa permasalahan teknis menjadi kendala dari sistem AMPS ini sehingga sistem ini tidak berkembang dan bahkan ditinggalkan setelah teknologi digital berkembang.

B. GSM (*Global System for Mobile telecommunication*)

GSM merupakan generasi kedua setelah AMPS, GSM pertama kali dikeluarkan pada tahun 1991 dan mulai berkembang pada tahun 1993 dengan diadopsi oleh beberapa negara seperti Afrika Selatan, Australia, Timur Tengah, dan Amerika Utara. Perkembangan pesat dari GSM disebabkan karena penggunaan system yang digital sehingga memungkinkan pengembang untuk mengeksploitasi penggunaan algoritma dan digital serta memungkinkannya penggunaan *Very Large Scale Intergration* (VLSI). Untuk mengurangi dan memperkecil biaya Handled terminalnya, pada saat ini GSM telah menggunakan fitur *Intelegent Network* (jaringan kecerdasan).

GSM adalah system telekomunikasi bergerak dengan menggunakan sistem selular digital. GSM pertama kali dibuat memang dipersiapkan untuk menjadi sistem telekomunikasi bergerak yang memiliki cakupan internasional berdasarkan pada teknologi *Multyplexing Time Division Multiple Access* (TDMA). GSM mempunyai frekuensi 900 Mhz selain itu GSM juga menggunakan frekuensi 1800 Mhz dengan nama *Personal Communication Network*. GSM juga menyediakan layanan untuk mengirimkan data dengan kecepatan tinggi yang menggunakan teknologi *High Speed Circuit Switch Data* (HSCSD) yang mampu mengirimkan data sampai 64 Kbps hingga 100 Kbps. Di Indonesia jaringan GSM di tempati oleh PT. Telkomsel, Exelkomindo, Satelindo, Indosat.

C. CDMA (*Code Division Multiple Access*)

CDMA merupakan generasi ketiga (3G). teknologi telpon tanpa kabel sangat dirasakan perkembangannya, dengan munculnya berbagai macam jenis telepon selular. Sekarang ini yang sedang berkembang adalah telepon tanpa kabel yang menggunakan *Code Devision Multiple Access* yang menggunakan teknik penyebaran spectrum. Berbeda dengan metode *Global System for Mobile Communication* (GSM) yang menggunakan *Time Division Multiplexing* (TDM), CDMA tidak memberikan penanda pada frekuensi khusus pada setiap user. Setiap channel menggunakan spectrum yang tersedia secara penuh. Percakapan individual akan di encode atau di sandikan dengan pengaturan digital secara pseudo random. CDMA merupakan perkembangan AMPS yang pertama kali digunakan oleh militer Amerika Serikat sebagai komunikasi Intelejen pada waktu perang. Perkembangan CDMA tidak secepat perkembangan GSM yang banyak diadopsi oleh sebagian besar operator di berbagai macam negara. Di Indonesia untuk jaringan CDMA ditempati oleh PT. Mobile-8, Telecom, Telkomflexy dan Esia.

2.5.1 Teori Dasar Layanan Pesan Multimedia (MMS)

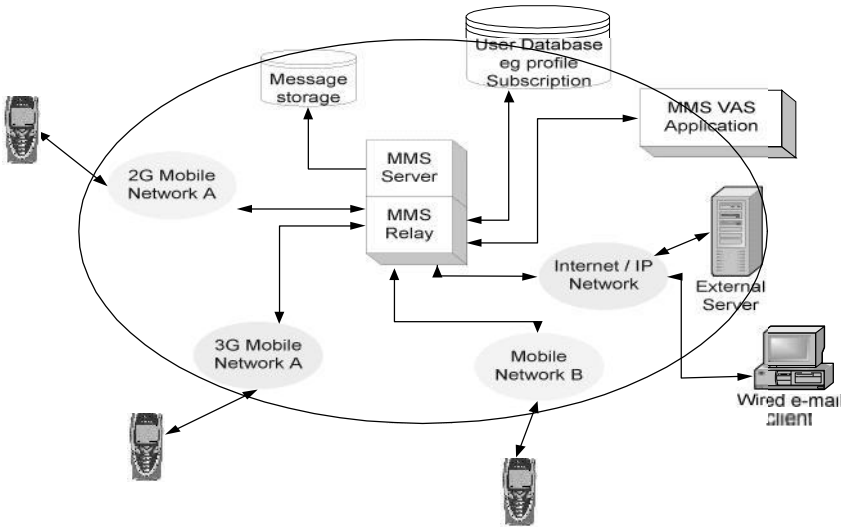
Layanan pesan multimedia (MMS/*Multimedia Messaging Service*) merupakan puncak dari evolusi layanan pesan singkat (SMS/*Short Messaging Service*) dan EMS (*Enhanced Messaging Service*). MMS menawarkan

perkembangan aplikasi secara menyeluruh sehingga pesan yang dikirim lebih kreatif dan menghibur. Pesan MMS dapat berupa teks, grafik/citra, data, animasi, audio, dan video.

Tidak seperti SMS yang menggunakan sinyal *link* dengan kapasitas minim, MMS menggunakan *main data channel* seperti GPRS (*General Packet Radio Service*). Pesan disimpan sementara pada MMSC jika pemakai tidak segera men-*download*. Pada MMS terdapat fitur *user profile* dan konversi data yang bergantung pada spesifikasi alat komunikasi (telepon selular) dan pemakai. *User profile* penting untuk mengetahui waktu ketika pesan di-*download*, karena pesan MMS terlalu besar sehingga tidak bisa segera sampai ke *receiver*. Kapasitas rata-rata SMS sekitar 140 byte. Sedangkan kapasitas rata-rata pada MMS sekitar 30.000 byte bahkan bisa mencapai 100.000 byte. Pesan yang dikirim tergantung kemampuan jaringan selular dan telepon selular.

A. Arsitektur MMS

Gambaran umum arsitektur MMS, MMSE (*MMS Environment*) meliputi seluruh elemen layanan yang dibutuhkan untuk mengirim, menyimpan dan pemberitahuan (*notification*). MMSE dapat diletakkan dalam satu jaringan atau terdistribusi dalam beberapa jaringan berbeda. MMS menggunakan teknologi WAP untuk komunikasi selular. Jaringan MMS dibangun pada arsitektur WAP, dalam hal ini WAP *gateway* menyediakan akses standar fasilitas WAP seperti HTTP, OTA dan kemampuan lainnya. Pesan multimedia dikirim oleh WSP (*WAP Session Protocol*) dan HTTP.



Gambar 2.9 Arsitektur MMS

Koneksi antar tipe jaringan yang berbeda dilakukan oleh IP (*Internet Protocol*) dan juga dilengkapi *messaging protocol*. *MMS server* bertanggung jawab menyimpan dan menangani pesan yang masuk dan keluar, mengatur aliran pesan multimedia dari dan ke telepon selular, dan telepon selular ke internet. Sebaliknya, *MMS server* juga menyediakan media penyimpanan dan operasional yang mendukung pesan multimedia. Tergabung dengan *MMS server* adalah *MMS proxy relay*, yang bertanggung jawab mengirimkan pesan antar *messaging* sistem yang berbeda, menentukan nilai data (*Call Detail Record*), dan mengidentifikasi kemampuan terminal telepon selular penerima. *MMS proxy relay* juga bertanggung jawab mengkonversi pesan MMS yang disesuaikan menurut kemampuan telepon selular penerima dan menjaga kompabilitasnya. Misalnya jika sebuah terminal MMS mengirim pesan beresolusi warna tinggi ke terminal MMS yang hanya mendukung warna hitam-putih, resolusi citra rendah, MMSC akan mengkonversi gambar tersebut kedalam bentuk hitam-putih. Fungsi ini juga diterapkan pada video klip, gambar dan file-file audio. MMS *proxy relay* berkaitan dengan aplikasi yang dijalankan pada MMS untuk mengembangkan berbagai aspek layanan seperti *store and forward*, menjamin pengiriman, data pelanggan, *operator constraint* dan informasi tagihan.

Basis data pemakai MMS terdiri dari satu atau lebih entitas yang berisi informasi pemakai, seperti langganan dan konfigurasi (contohnya profil pemakai dan lokasinya). *MMS user agent* berfungsi sebagai *application layer* yang terletak pada telepon selular atau alat komunikasi lain yang mampu menampilkan, menggabungkan dan menangani (mengirim, menerima, menghapus, dan lainnya) pesan MMS.

File yang akan dikirimkan (baik itu berupa teks, gambar, maupun audio dan video) akan dikonversikan terlebih dahulu pada telepon selular ke dalam format file *.mpr* yang merupakan format file standar dari MMS.

2.5.2. Android

Android merupakan sebuah sistem sistem operasi untuk perangkat *mobile* berbasis *linux* yang meliputi sistem operasi, *middleware*, dan aplikasi yang dirilis oleh Google. Sedangkan Android SDK (*Software Development kit*) menyediakan Tools dan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java (Mulyadi, 2010). Android dikembangkan oleh Google bersama OHA (*Open Handset Alliance*) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware, Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

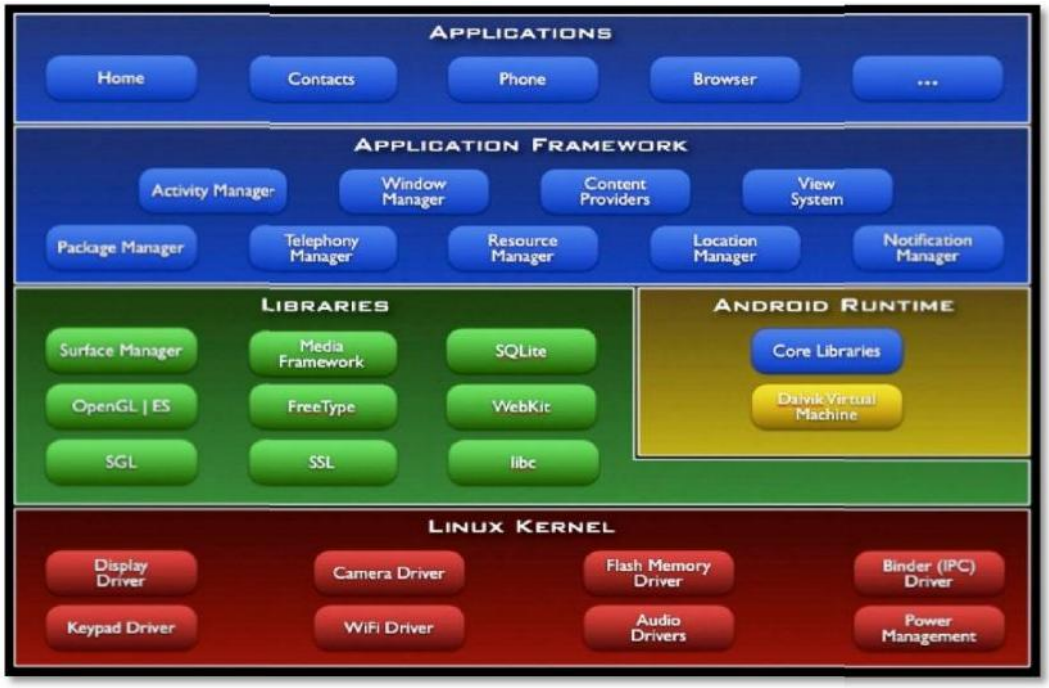
A. Sejarah Android

Pada tahun 2005 Google mengakusisi Android Inc yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Kemudian pada tahun itu juga memulai membangun *platform* Android secara intensif. Kemudian pada tanggal 5 November 2007 Android pertama kali diluncurkan, dan *smartphone* pertama yang menggunakan sistem operasi Android dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini Android telah merilis beberapa versi Android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa

versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama Cupcake, versi 1.6 dirilis pada 15 September 2009 diberi nama Donut, versi 2.0/2.1 dirilis pada 26 Oktober 2009 diberi nama Éclair, versi 2.2 dirilis pada bulan Mei 2010 diberi nama Froyo dan versi 2.3 dirilis pada Desember 2010 yang diberi nama Gingerbread.

B. Arsitektur Android

Dalam paket sistem operasi Android terdiri dari beberapa unsur seperti tampak pada gambar 2.1. Secara sederhana arsitektur Android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.10 Arsitektur Android (*sumber: http://developer.*Android*.com*)

C. Linux Kernel

Android bukan *linux*, akan tetapi Android dibangun diatas *linux kernel* versi 2.6 yang kehandalannya sudah teruji. Untuk inti sistem servis *linux* yang digunakan seperti keamanan, manajemen momori, proses manajemen, *network* dan *driver* model. Seperti yang terlihat di gambar 2.6, linux kernel menyediakan *Driver* layar, *driver* kamera, flash memori, IPC (*Interprocces Communication*) untuk mengatur aplikasi dan keamanan, *driver* keypad, *driver* WiFi, *driver* audio,

dan power management. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software* stacknya (http://developer.Android.com).

D. *Libraries*

Android menyertakan linbraries C / C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *developer* aplikasi melalui *framework* aplikasi Android. Beberapa inti libraries tercantun dibawah ini (Mulyadi, 2010):

1. *System C library*

   Variasi dari implementasi BSD (*Barkeley Software Distribution*) berasal dari sistem standar C *library* (*libc*), sesuai untuk perangkat *embedded* berbasis linux.

2. *Media libraries*

   Untuk merekam dan memutar berbagai format *Audio* dan *Video.*

3. *Surface Manager*

   Mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.

4. *LibWebCore*

   Library untuk mesin *web* pada *browser* Android.

5. *SGL*

   Merupakan library untuk proses mesin grafis 2D.

6. *3D libraries*

   Digunakan untuk proses gambar yang membutuhkan daya 3D.

7. *FreeType*

   Merupakan *library* untuk bitmap dan *vektor font rendering*

8. *SQLite*

   Merupakan *library* mesin *database* yang kuat dan ringan, dan peghubung aplikasi yang tersedia.

E. **Android *Runtime***

Pada Android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi Android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android *Runtime*

merupakan mesin virtual yang membuat aplikasi Android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android *Runtime* terdapat 2 bagian utama,diantaranya (http://deveoper.Android.com):

Pustaka Inti, Android dikembangkan melalui bahasa pemrograman Java, tapi Android *Runtime* bukanlah mesin *virtual* Java. Pustaka inti Android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus Android.

Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan diatas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

F. Applications Frameworks

Android merupakan "*Open Development Platform*" dimana Android menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangakat keras, akses informasi *resources,* menjalankan *service background*, mengatur alarm, menambah status pemberitahuan dan lainnya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti.

Applications Frameworks merupakan layer dimana para pembuat aplikasi melakukan pengembangan aplikasi yang akan dijalankan di sistem operasi Android. Komponen-komponen yang termasuk dalam *Applications Frameworks* adalah sebagai berikut (http://deveoper.Android.com):

a. *Views*
b. *Content Provider*
c. *Resource Manager*
d. *Notification Manager*
e. *Activity Manager*
f. *Applications* dan *Widgets*

Applications dan *Widgets* ini adalah layer yang berhubungan dengan aplikasi saja. Biasanya aplikasi di *download* kemudian dilakukan instalasi dan jalankan aplikasi tersebut. Di layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain.

2.5.3. Android SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Untuk sumber SDK Android dapat dilihat dan diunduh langsung kesitus resmi Android di http://developer.Android.com.

A. Komponen Aplikasi

Aplikasi Android ditulis dalam bahasa pemograman Java. Kode Java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, prosesnya di paket oleh *tools* yang dinamakan *atp tools* kedalam paket Android, sehingga menghasilkan file dengan ekstensi *apk*. Ada 4 jenis komponen pada aplikasi Android, yaitu (http://deveoper.Android.com):

1. *Activities*

Suatu *Activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Pada umumnya sebuah aplikasi Android memiliki banyak *activity*, tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain, kita dapat melakukan dengan satu even, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah windows *activity* dinyatakan dengan *method Activity*.setContentView(). ContentView adalah objek yang berada pada *root hirarki.*

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*. Komponen *service* diproses tidak terlihat,

memperbaharui seumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang terus diproses, bahkan ketika *activity* tidak aktif.

3. *Broadcast Receiver*

Broadcasr receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast receiver* tidak memiliki *user interface*, tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima atau menggunakan *notification manager* untuk memberi tahu pengguna, seperti lampu latar atau getaran.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi dari *content provider*.

B. Kelebihan *Platform* Android

Persaingan *platform* atau sistem operasi semankin ketat, ini dapat dilihat dari banyaknya sistem operasi yang ada seperti, Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut (Amiral, 2010) :

Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royality*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dantanpa terikat oleh vendor manapun.

Arsitektur komponen dasar Android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya,

bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.

Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database* SQL, *browser* dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.

Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.

Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun *game* yang berbeda.

Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.6. Analisa dan Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata (A.Suhendar, 2002).

Perusahaan *software*, Rational *Software*, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysis Design* (O*OAD).*

2.6.1. *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti dan Wahono, 2006).

Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain: *usecase diagram, class diagram, statechart diagram, activity diagram, sequence diagram, collaboration diagram, component diagram, deployment diagram.*

A. *Usecase Diagram*

Usecase diagram merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah *usecase* merepresentasikan interaksi antara aktor dengan sistem. *Usecase* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu Dharwiyanti (2006).

Dalam sebuah sistem *usecase diagram* akan sangat membantu dalam hal menyusun *requirment*, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem.

B. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek danmerupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan(atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) Dharwiyanti (2006).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu, nama, *stereotype* ,atribut dan metoda.

C. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan Dharwiyanti (2006).

D. *Statechart Diagram*

(Suhendar dkk, 2002) menyebutkan bahwa *statechart diagram* digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart diagram* memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas kepada yang lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas. *Statechart diagram* khususnya digunakan untuk memodelkan tahap-tahap diskrit dari sebuah siklus hidup objek, sedangkan *activity diagram* lebih cocok digunakan untuk memodelkan urutan aktivitas dalam sustu proses.

E. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini Dharwiyanti (2006).

F. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang,bagaimana masing-masing alir berawal, *decision* yang

mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi padabeberapa eksekusi Dharwiyanti (2006).

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

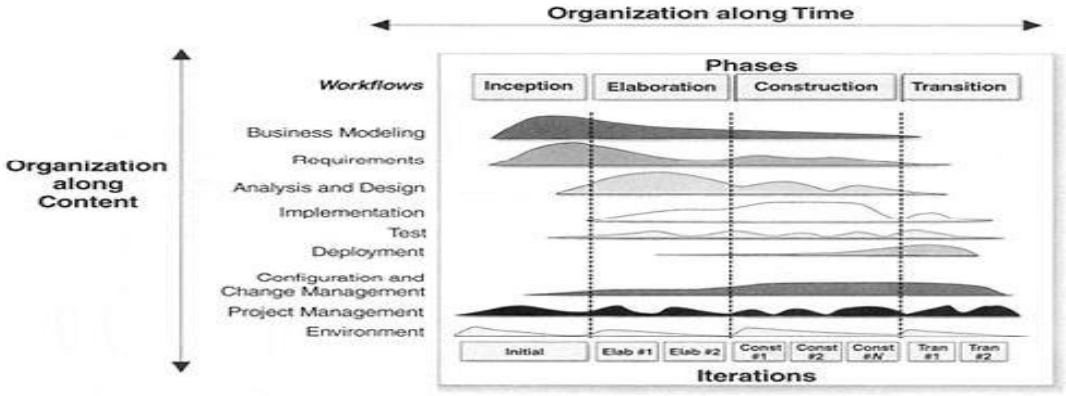
2.6.2. *Rational Unified Process* (RUP)

Rational Unified Process adalah sebuah Proses Rekayasa Perangkat Lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk memastikan perangkat lunak yang berkualitas tinggi dan sesuai kebutuhan penggunanya dalam anggaran dan jadwal yang dapat diprediksi (Kruchten, 2000).

RUP mengarahkan kita terhadap pengembangan perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain (Kruchten, 2000):

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.
2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures* untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*,
6. *Control changes to software.* kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar 2.6 dibawah dapat dilihat bahwa RUP memiliki 2 dimensi, yaitu:



Gambar 2.11 Struktur Proses Dua Dimensi RUP (sumber: Kruchten, 2000)

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari fase selanjutnya. Setiap fase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction,* dan *Transition.*

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how* dan *when*. Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Manegement, Project Management, Environtment.*

A. Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception, Elaboration, Construction* dan *Transition* (Rational Team, 2001).

1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi dan analisa masalah, untuk itu diperlukan juga identifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *usecase* dan gambarannya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*.

Hasil yang diperoleh pada fase ini adalah :

a. Dokumen visi (visi dari kebutuhan projek, kata kunci, batasan utama).
b. Daftar kata.
c. *Business case,* termasuk didalamnya konteks bisnis, kriteria sukses, pengenalan pasar dan proyeksi keuangan.
d. Inisialisasi penilaian resiko.
e. Rencana proyek dan menunjukan fase serta iterasi.
f. Model bisnis jika diperlukan

Kriteria evaluasi untuk fase *Inception* adalah :

a. Menyesuaikan *stakeholder* dengan *scope definition* dan perkiraan biaya atau perkiraan jadwal.
b. Pemahaman terhadap *usecase* utama
c. Kredibilitas dari perkiraan biaya, jadwal, prioritas, resiko dan proses pengembangan.
d. Pemahaman terhadap *prototype*

2. Fase *Elaboration*

Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-unsur yang memiliki resiko besar terhadap proyek. Adapun hasil dari fase elaboration adalah:

a. *Usecase* model, seluruh *usecase* dan aktor telah teridentifikasi.
b. *Requirement* tambahan yang mungkin tidak bersifat fungsional bagi proyek.

c. *Software Architecture Description* (Deskripsi Arsitektur Perangkat Lunak).
d. Prototipe dari arsitektur yang dapat dieksekusi.
e. Revisi daftar tingkat resiko dan revisi *business-case*.
f. Rencana pengembangan keseluruhan proyek.
g. Persiapan dokumen panduan bagi pengguna (*user manual*).

Kriteria utama dalam fase elaboration melibatkan pertanyaan berikut :

a. Apakah produk sudah stabil ?
b. Apakah rancangan arsitekturalnya sudah stabil ?
c. Apakah saat demo prototipe, unsur yang memilki resiko telah bisa di atur ?
d. Apakah rencana kontruksi telah detail dan akurat ?
e. Apakah *stakeholder* bersedia dan menyepakati visi dari pengembangan proyek tersebut?
f. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan dapat diterima?

3. Fase *Contruction*

Selama fase kontruksi, semua komponen dan fitur yang dikembangkan terintergrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase *Inception* dan *Elaboration*, menjadi pengembangan *deplyoable product*. Kriteria evaluasi terhadap fase *Construction* ini adalah :

a. Apakah peluncuran produk cukup baik dan dapat diterima di komunitas pengguna?
b. Apakah semua *stakeholder* siap untuk beralih ke komunitas pengguna?
c. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk. Kriteria evaluasi untuk fase *Transition* adalah :

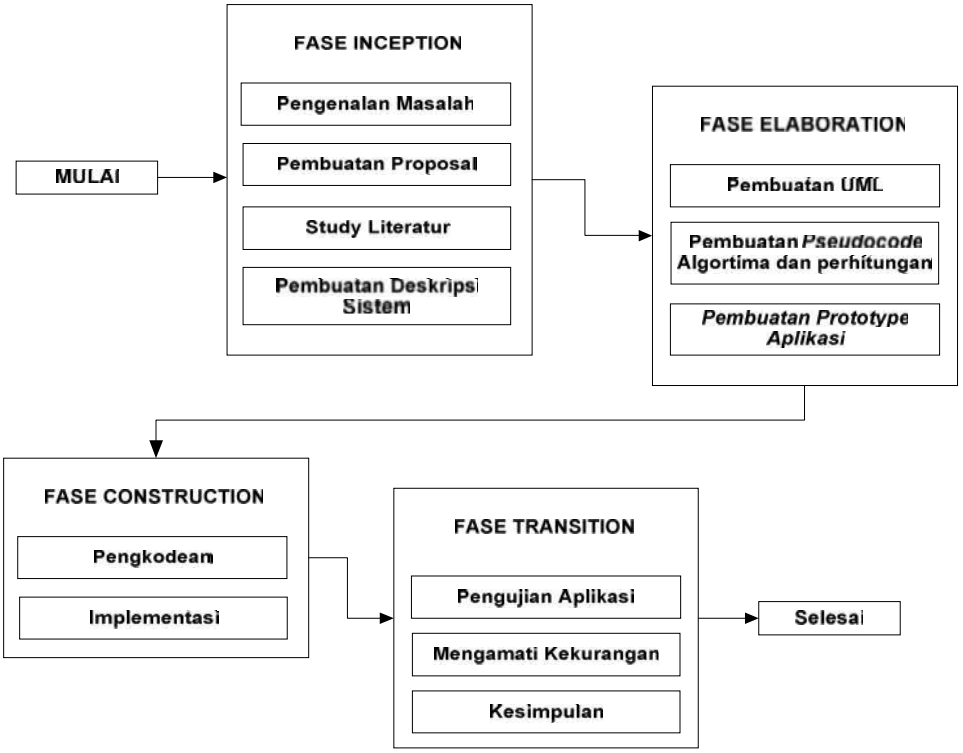
a. Apakah pengguna merasa puas?
b. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

BAB III

METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang langkah-langkah yang digunakan untuk membahas permasalahan yang diambil dalam penelitian. Tahapan Penelitian yang akan dilaksanakan pada pembuatan aplikasi steganografi dengan menggunakan DCT dan kriptografi RC6 pada Android dengan menggunakan RUP (*Rational Unified Process*):

3.1. Alur tahapan RUP



Gambar 3.1 Tahapan Penelitian dengan Metode RUP

Alur tahapan RUP yang akan digunakan dalam membuat rancang bangun aplikasi pengamanan pengiriman pesan dalam gambar menggunakan teknik steganografi dengan Metode DCT dan kriptografi RC6 ini dapat dilihat pada gambar 3.1 dengan penjelasan tiap fase sebagai berikut:

3.1.1. Fase Inception

Fase ini merupakan fase analisa, dimana dilakukan beberapa kegiatan untuk mengidentifikasi aplikasi yang akan dibuat. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

a. Pengenalan masalah, memahami permasalahan yang terjadi, mengapa diperlukan suatu pengamanan dalam pertukaran informasi *digital* yang digunakan oleh setiap orang yaitu MMS (*Multimedia Message Service*) sehingga dibutuhkan suatu teknik steganografi dalam mengamankan pesan yang dikirim pada perangkat *smartphone* yang bersistem operasi android.
b. Pembuatan proposal, yaitu mencakup latar belakang permasalahan, rumusan masalah, batasan penelitian, tujuan, manfaat, sistematika penulisa, landasan teori, dan metodologi penelitian.
c. Studi Literatur, mencakup penelusuran teori-teori yang berhubungan dengan permasalahan, yang bersumber dari buku, jurnal, artikel internet dan penelitian-penelitian sejenis yang dapat mendukung pemecahan masalah dalam penelitian yang dilakukan.

3.1.2 Fase *Elaboration*

Fase ini merupakan fase perancangan desain aplikasi yang akan dibangun, sesuai dengan hasil analisa pada fase sebelumnya. Pada fase *elaboration* akan dilakukan tugas-tugas sebagai berikut :

a. Pembuatan UML yang meliputi *usecase diagram, class diagram, activity diagram, sequence diagram* dan *deployment diagram.*
b. Pembuatan *Pseudocode* Algoritma dan perhitungan manual.
    i. Algoritma matriks 8x8
       Mengubah suatu matriks gambar asli dengan matriks gambar 8x8
    ii. Algoritma kompresi DCT (*transformasi*)
    iii. Algoritma kuantisasi
    iv. Algoritma *entropy coder*
       Bertujuan untuk menyimpan matriks dalam urutan zig - zag
    v. Algortima *entropy decoder*

vi. Algoritma dekuantisasi

vii. Algoritma dekompresi DCT

viii. Pembangkit sub kunci

Kunci dari pengguna ini akan dimasukkan oleh pengguna pada saat akan melakukan proses enkripsi dan dekripsi. Kunci ini memiliki tipe data *string* dan memiliki panjang 16 *byte* (16 karakter)

ix. Baca Masukan untuk Proses Enkripsi

Yang dilakukan pada tahapan ini adalah membaca teks yang menjadi masukan pada proses enkripsi, yaitu *field* dari aplikasi enkripsi MMS. Pada proses enkripsi pesan, *field* nya adalah isi pesan.

x. Enkripsi meliputi *whitening* awal, iterasi, dan *whitening* akhir.

xi. Baca File Masukan untuk Proses Dekripsi

Yang dilakukan pada tahapan ini adalah membaca teks yang menjadi masukkan pada proses dekripsi, yaitu *record* dari hasil pesan yang telah dienkripsi pada pengirim dan menjadi *field* pesan pada penerima.

c. Pembuatan Prototype Antarmuka Aplikasi.

3.1.3 Fase *Construction*

Fase ini meliputi kegiatan pengkodean dan implementasi. Fase ini dilakukan setelah fase *elaboration* selesai dilakukan, karena fase *construction* bisa dilaksanakan setelah fase sebelumnya selesai dilakukan. Fase ini adalah fase dimana pembuat aplikasi mulai membangun aplikasi berdasarkan hasil fase *inception* dan fase *elaboration.*

3.1.4 Fase *Transition*

Setelah menyelesaikan fase *construction*, sehingga kemudian beralih ke fase terakhir yaitu fase *transition.* Fase ini merupakan fase dimana akan dilakukan *deploying* dan pengujian aplikasi untuk melihat kekurangan aplikasi oleh pembuat aplikasi, dan testing tiap fungsi pada aplikasi, kemudian diambil kesimpulan tentang penggunaan aplikasi.

BAB IV

ANALISA DAN PERANCANGAN

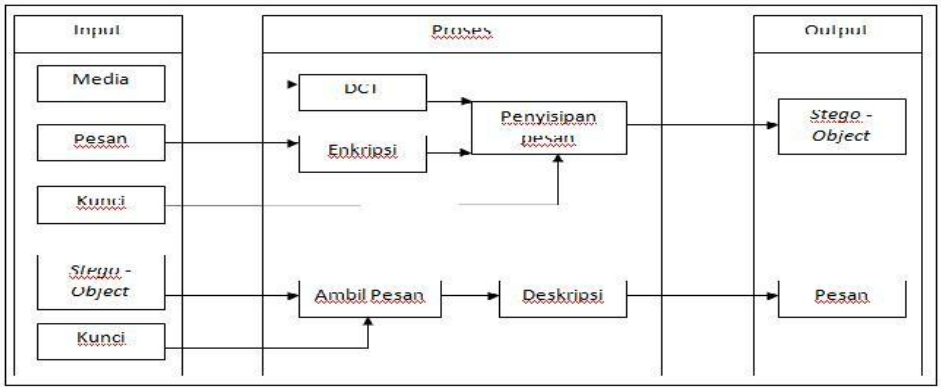
4.1 Fase *Inception*

Berdasarkan metodologi penelitian dalam pembuatan Tugas Akhir ini yang menggunakan RUP (*Rational Unified Process*) maka alur pertama yang harus dilakukan dari tahapan RUP yaitu *inception* dimana dalam bab sebelumnya telah dijelasakan poin-poin yang harus dilakukan dalam fase *inception* ini.

Sebagian poin-poin yang termasuk dalam fase *inception* sudah dimasukkan didalam Bab I sampai Bab II yang meliputi pengenalan masalah, pembuatan proposal, dan studi litetarur. Point berikutnya untuk pembuatan deskripsi sistem akan dijelaskan pada Bab ini yang meliputi Deskripsi Umum Sistem dan Deskripsi Umum Analisa Struktur DCT dan RC6.

4.1.1 Deskripsi Umum Sistem

Pada dasarnya sebuah aplikasi steganografi memiliki proses yang dapat merahasiakan pesan (proses *embedding*) dan cara untuk mengambil kembali pesan yang telah disembunyikan (proses *retrieving*). Begitu juga dengan rancang bangun aplikasi steganografi untuk penyisipan *text* ke dalam *image* dengan metode *Discrete Cosine Transform* (DCT) yang dibahas dalam laporan ini akan memiliki dua proses utama steganografi tersebut. Kebutuhan akan data-data inputan *(requirement data)* pada sebuah aplikasi steganografi seperti yang telah dijelaskan pada landasan teori juga menjadi kebutuhan utama pada rancang bangun aplikasi steganografi ini. Tidak lupa juga sebuah kunci yang berfungsi untuk mengamankan pesan didalam data – data inputan. Sistem ini meliputi gambar berformat JPEG sebagai media penyisipan, teks sebagai pesan yang ingin disisipkan, serta kunci sebagai pengaman yang menggunakan RC6. Untuk lebih jelasnya dapat dilihat pada gambar 4.1 berikut.



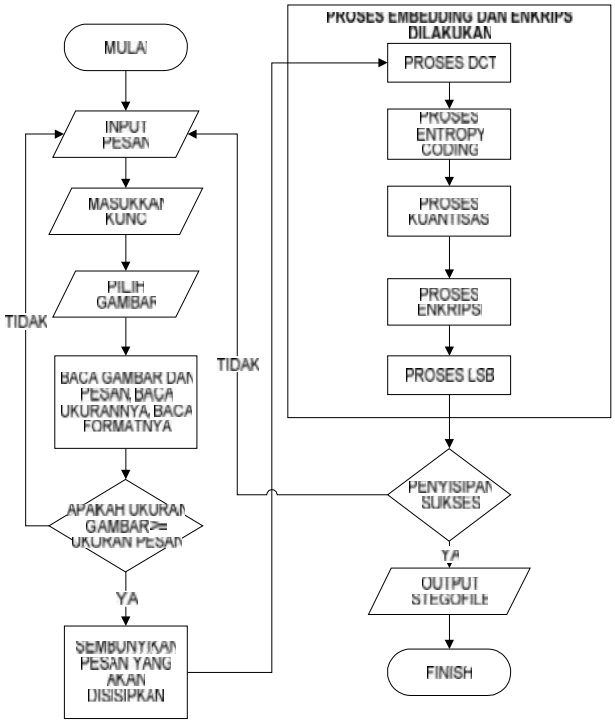
Gambar 4.1 Gambaran Umum Sistem

Berdasarkan gambar 4.1 dapat diketahui bahwa rancang bangun aplikasi steganografi ini memiliki proses, kebutuhan inputan data, dan output yang sama seperti aplikasi steganografi pada umumnya:

1. Input merupakan kebutuhan inputan data yang dibutuhkan oleh sistem yang terdiri dari empat jenis data yang dibedakan berdasarkan proses-proses utama *(main process)* steganografi, yaitu pesan rahasia *(message)* yang akan disisipkan, media penampung *(carrier file)* yang menyatakan tempat dimana pesan rahasia akan disisipkan untuk proses *embedding*, dan kunci yang digunakan untuk menampilkan pesan tersembunyi, serta data yang telah membawa pesan *(stego-object)* untuk proses pengambilan pesan (*retrieving).*
2. Proses, menyatakan proses-proses utama yang terdapat pada rancang bangun aplikasi steganografi, yaitu proses penyembunyian atau penyisipan *(embedding)* dan proses pengambilan kembali pesan *(retrieving).* Pada proses inilah penerapan metode DCT itu terjadi yang dimulai dengan mengonversi *requirement data* dalam biner, kemudian dilakukan penyisipan bit-bit pesan ke bit-bit LSB medianya.
3. *Output* merupakan hasil dari *main process* yang terjadi pada sistem. Output dari e*mbedding* disebut dengan *stegofile*, sementara output dari *retrieving* adalah pesan rahasia yang tersembunyi di dalam *stegofile.*

4.1.2. Proses Penyembunyian Pesan (*Embedding*)

Proses ini berfungsi untuk menyembunyikan pesan rahasia ke dalam media *inputan*. Secara detil, proses ini ditunjukan pada gambar 4.2 dibawah ini.



Gambar 4.2 *Flowchart Embedding* dan Enkripsi Pesan

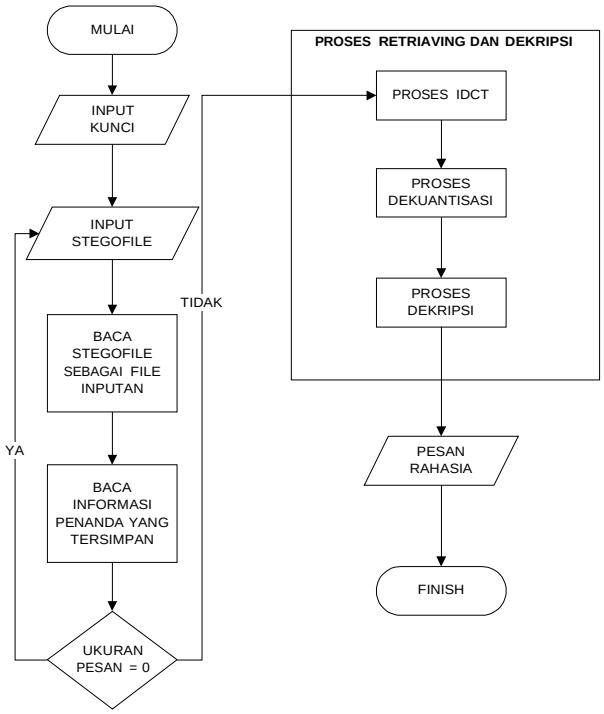
Berdasarkan *flowchart* proses penyisipan yang ditunjukkan pada gambar 4.2 diatas dapat dijelaskan bahwa:

1. Penyisipan pesan rahasia dimulai dengan menginputkan *requirement data* ke dalam sistem, yang terdiri dari nomor kunci rahasia untuk mengacak dan mengembalikan pesan, pesan rahasia yang akan disembunyikan, dan media gambar yang berupa JPEG.
2. Selanjutnya, aplikasi akan membaca pesan rahasia dan ukurannya, kemudian membaca media penyisipan yang berupa JPEG. Dilakukan validasi terhadap ukuran media penyisipan dan pesan yang akan disembunyikan. Jika ukuran media penyisipan lebih besar atau sama dengan ukuran pesan rahasia, maka proses akan berlanjut. Namun, jika ukuran media penyisipan lebih kecil dari ukuran pesan rahasia atau pesan lebih besar dari medianya, maka proses akan kembali ke langkah pertama.

3. Sembunyikan informasi penanda pesan yang akan disisipkan pada media. Kalau tipe pesan yang disisipkan adalah teks, maka yang dijadikan penanda adanya pesan di dalam sebuah media adalah hanya berupa ukuran dari pesan yang akan disisipkan. Bagian ini menjadi bagian yang sangat penting ketika penerima ingin mengambil *(retrieving)* pesan.
4. Diperoleh output berupa *stegofile* (*file* baru yang telah membawa pesan).
5. Selesai.

4.1.3. Proses Ekstrasi Pesan (*Retrieving*)

Proses ini berfungsi untuk memperoleh kembali pesan rahasia yang telah disembunyikan. Secara detil, proses ini ditunjukan pada gambar 4.3 dibawah ini.



Gambar 4.3 *Flowchart Retrieving* dan Dekripsi Pesan

Berdasarkan alur proses *retrieving* yang ditunjukkan pada gambar 4.3 dapat dijelaskan bahwa:

1. *Retrieving* terjadi apabila *requirement data* berupa *stegofile* dan kunci telah diinputkan pada sistem.
2. Sistem akan membaca informasi penanda dari *stegofile* yang diinputkan.

3. Membaca barisan bit-bit *stegofile* untuk menemukan informasi ukuran pesan yang tersimpan. Jika ukuran pesan berubah dengan nilai aslinya, maka ditemukan informasi ukuran pesan yang tersimpan pada *stegofile*
4. Output yang dihasilkan adalah pesan rahasia.
5. Selesai.

4.1.4 Perubahan Format Inputan Gambar Dari JPEG Ke PNG

Format inputan gambar untuk aplikasi ini menggunakan JPEG namun outputnya dibuat menjadi PNG dikarenakan setiap kali menyimpan format dalam bentuk JPEG dari tipe lain ukuran gambar biasanya mengecil, kualitasnya menurun dan tidak dapat dikembalikan lagi. Namun kelebihan dari file PNG bersifat lossless compression yaitu tidak menghilang kan data yang telah dikompresi oleh file JPEG. Format file PNG ini juga banyak digunakan dari format file citra yang lain pada mobile maupun dalam pengolahan citra.

4.2 Fase *Elaboration*

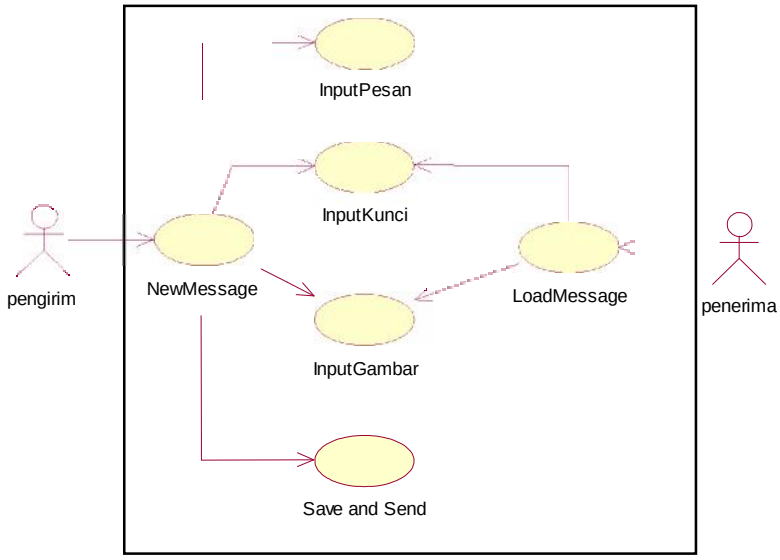
Tahap kedua dalam alur proses RUP yaitu *elaboration* dimana dalam alur ini dilakukan perancangan terhadap aplikasi yang akan dibuat. Dalam perancangan yang akan dibuat meliputi pembuatan UML, perancangan *pseudocode* algoritma dan perhitungannya serta pembuatan *prototype* atau perancangan *interface* aplikasi.

4.2.1 Perancangan Sistem

Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan sistem steganografi dengan DCT dan kriptografi RC6 ini. Perancangan-perancangan yang akan dijelaskan dalam laporan ini meliputi perancangan model dalam bentuk UML (*Unifed Modeling Language*) yang terdiri dari *Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram, Deploy Diagram.*

4.2.1.1 Model *Use Case*

Hal-hal yang dapat dilakukan oleh pengguna terhadap sistem dapat dilihat pada diagram *use case* pada Gambar 4.2 berikut



Gambar 4.4 Use Case Diagram

Perangkat lunak ini memiliki 6 buah use case dan 2 buah aktor. Pengirim merupakan pengguna yang melakukan penyisipan pesan ke dalam gambar, sedangkan penerima adalah pengguna yang melakukan ekstrasi pesan. Adapun penjelasan dari setiap *use case* diagram dari gambar 4.2 dijelaskan pada tabel dibawah ini.

Tabel 4.1 Spesifikasi Use Case New Message

AKTOR UTAMA	Pengirim
KONDISI AWAL	-
KONDISI AKHIR	Sistem akan memvalidasi format gambar yang akan disisipi pesan
MAIN SUCCES SCENARIO	1. Dimulai ketika aktor memilih proses *New Message.* 2. Sistem akan menampilkan fungsi untuk memilih gambar dari *gallery* yang akan disisipi pesan. 3. Kebutuhan sistem adalah pesan yang akan

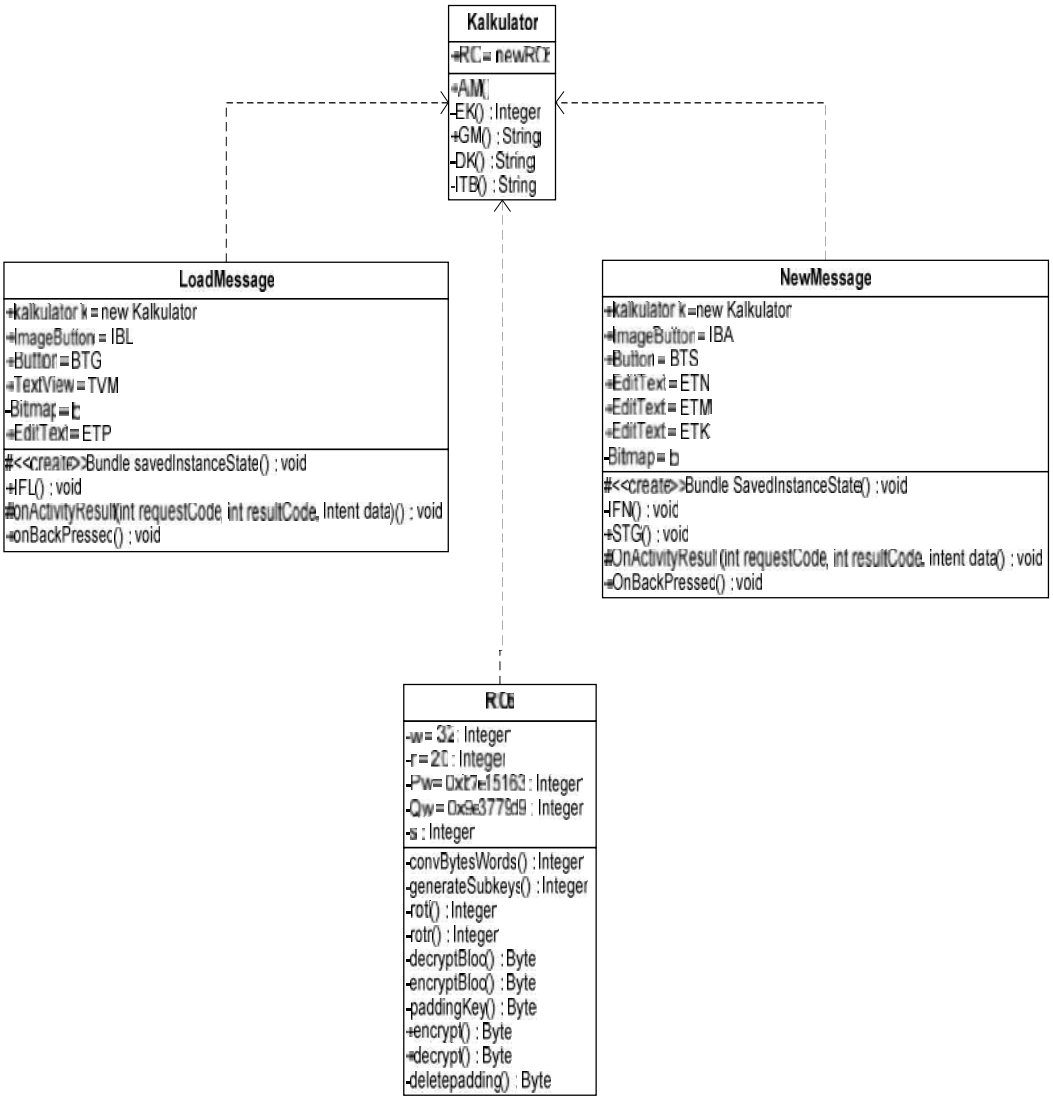
	disisipkan (*message*), kunci (*key*), media penampungnya (*carrier file*). 4. Sistem mevalidasi data yang terinput. 5. Jika *format* gambar yang terinput sesuai, maka sistem dapat melakukan proses *embedding*.
EXCEPTION	Jika yang terinput format yang salah, maka aplikasi akan "*Error*"

Tabel 4.2 Spesifikasi Use Case Load Message

AKTOR UTAMA	Penerima
KONDISI AWAL	-
KONDISI AKHIR	Data yang terinput adalah data *stegofile*
MAIN SUCCES SCENARIO	1. Dimulai ketika aktor memilih proses *Load Message.* 2. Sistem akan menampilkan fungsi untuk mengambil gambar dari *gallery* yang akan diambil kembali pesannya (*retriaving*). 3. Sistem mevalidasi data yang terinput. 4. Jika *stegofile* yang terinput, maka sistem dapat melakukan proses *retrieving*.
EXCEPTION	Jika yang terinput bukan stegofile, maka aplikasi akan "*Error*"

4.2.1.2 Class Diagram

Class diagram digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat dalam sistem tersebut. Berdasarkan class diagram yang telah ada maka aplikasi ini dibangun dalam empat kelas yaitu, kalkulator, RC6, NewMessage, dan LoadMessage. Adapun keterangan dari masing – masing kelas tersebut dapat dilihat pada tabel 4.2 dibawah ini.



Gambar 4.5 Class Diagram Steganografi DCT dan RC6

Tabel 4.3 Deskripsi Perancangan *Class Diagram*

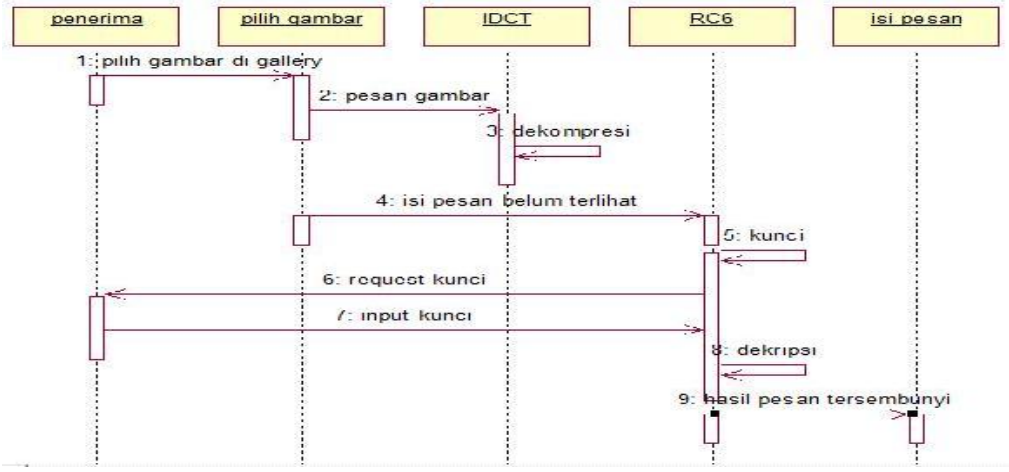
Nama Kelas	Nama File	Keterangan
Penerima	LoadMessage.java	Kelas ini merupakan tampilan utama dari aplikasi steganografi, ketika aplikasi dijalankan, maka kelas ini yang pertama dipanggil
Pengirim	NewMessage.java	Kelas ini merupakan tampilan untuk melakukan penulisan, pengenkripsian, dan pengririman pesan
Penyisipan	Kalkulator.java	Kelas ini merupakan kelas untuk menyisipkan pesan kedalam gambar
RC6	RC6.java	Kelas ini merupakan kelas algoritma enkripsi dekripsi dari RC6, serta algoritma dari penjadwalan kunci RC6

4.2.1.3 Sequence Diagram

Sequence Diagram adalah representasi dari interaksi-interaksi objek yang berjalan pada sistem. Dengan menggunakan sequence diagram kita dapat melihat bagaimana objek-objek bekerja. Sequence diagram dapat menampilkan bagaimana sistem merespon setiap kejadian atau permintaan dari user, dapat mempertahankan integritas internal, bagaimana data dipindah ke user interface dan bagaimana objek-objek diciptakan dan dimanipulasi.

Setiap sistem memiliki proses dan setiap proses memiliki dua kriteria, yaitu proses sederhana dan kompleks. Dengan demikian tidak seluruh proses pada sistem akan ditampilkan pada sequence diagram, melainkan hanya garis besarnya saja.

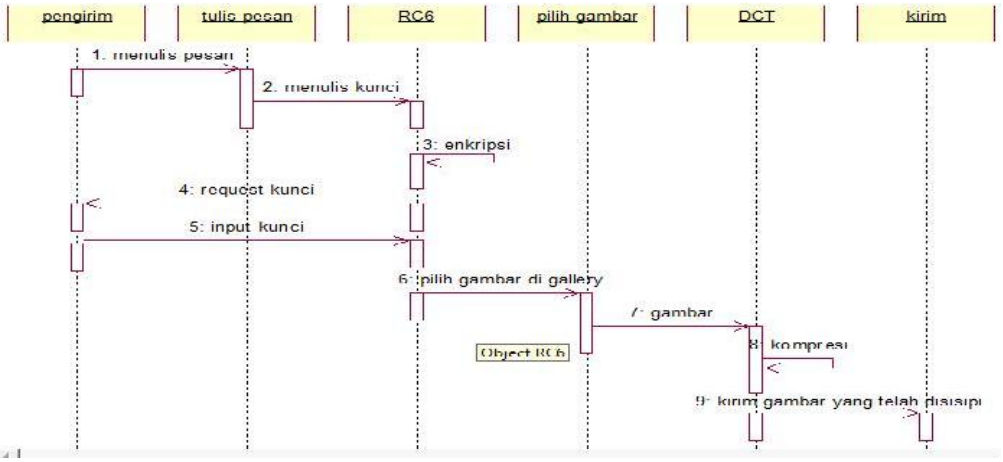
1. Sequence Diagram *Load Message*



Gambar 4.6 ***Sequence Diagram Load Message***

Dari Gambar diatas terlihat penerima memilih menu pilih gambar. Setelah menu pilih gambar di pilih, sistem akan mencari gambar yang berada di *memory* penyimpanan dan gambar tersebut akan mengalami proses IDCT yang akan mencari blok – blok yang telah disisipi pesan, proses selanjutnya adalah user diminta untuk memasuki kunci dekripsi pada fungsi RC6 dan mendekripsikan pesan yang tersisipi. Proses terakhir, pesan yang telah disisipi pada gambar telah berhasil dilihat atau dikembalikan.

2. Sequence diagram *New Message*



Gambar 4.7 ***Sequence Diagram New Message***

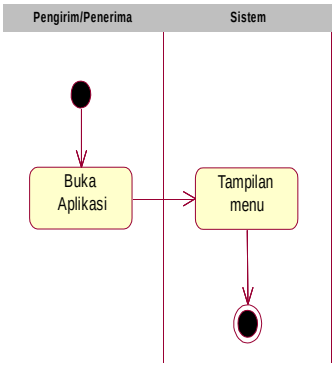
Dari gambar 4.8 diatas dapat dilihat bahwa pengirim menulis pesan yang akan disisipi ke dalam gambar, lalu memasukkan kunci untuk mengenkripsi pesannya dengan RC6. Kemudian user memilih gambar sebagai media penyisipan yang akan dikirim kepada penerima dengan menggunakan teknik DCT. Proses terakhir pengirim dapat memilih media untuk mengirimnya seperti, Bluetooth, MMS, Facebook, dan lain - lain.

4.2.1.4 Activity Diagram

Untuk memudahkan dalam perancangan *activity diagram* maka *activity diagram* dalam aplikasi ini akan dipecah menjadi beberapa bagian.

1. *Activity Diagram* Buka Aplikasi

Pada bagian pertama *activity diagram* seperti terlihat pada gambar 4.8 saat aplikasi dijalankan sistem akan memenggil *class* form utama untuk meletakkan aplikasi dalam melakukan inisialisai proses apa saja yang akan di *load* pertama kali, kemudian aplikasi akan menampilkan tampilan menu setelah tampilan menu utama tampil maka prosesakan selesai.

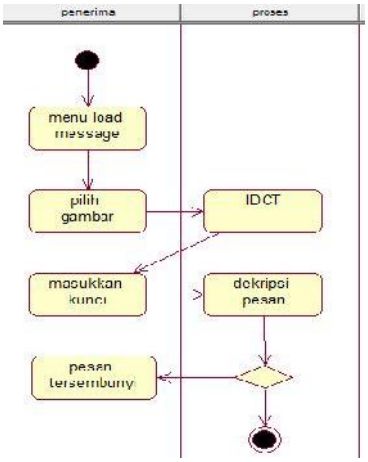


Gambar 4.8 *Activity Diagram* Buka Aplikasi

2. *Activity Diagram Load Message*

Activity diagram Load Message merupakan alur proses lanjutan dari *activity diagram* buka aplikasi. Alur kerja dari proses ini adalah setelah *user* masuk aplikasi kemudian user langsung masuk pada menu utama. Pertama user diharuskan memilih gambar pada gallery penyimpanan, gambar yang telah dipilih akan mengalami proses dekompresi agar mendapatkan blok – blok yang telah

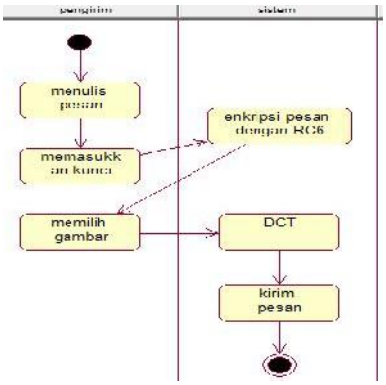
disisipi pesan, kemudian user memasukkan kunci. Setelah kuncinya diinputkan maka pesan yang telah tersisipi pada pesan akan didekripsi menggunakan RC6. Pesan yang sudah disisipi pada gambar akan muncul, jika kunci yang dimasukkan salah maka pesan yang muncul menjadi tidak beraturan.



Gambar 4.9 *Activity Diagram* Load Message

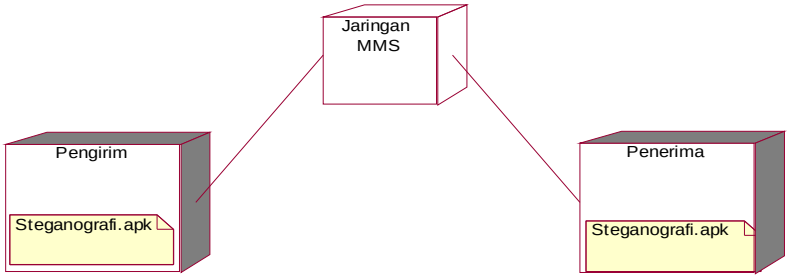
3. *Activity Diagram New Message*

Alur proses yang terjadi pada *activity diagram New Message* adalah setelah *user* memilih menu "kembali" pada menu *Load Message*, kemudian sistem akan menampilkan menu New Message. Pertama user harus menulis pesan yang akan di sisipkan ke dalam gambar. Kemudian user harus memasukkan kunci untuk menenkripsi pesan. Proses terakhir user harus memilih gambar yang berformat JPEG yang digunakan sebagai media penyisipannya, selanjutnya user bisa memilih mengirim gambarnya melalui media yang disediakan.



Gambar 4.10 *Activity Diagram New Message*

4.2.1.5. *Deploy Diagram*



Gambar 4.11 *Deploy Diagram* Stegano DCT & Kripto RC6

Deploy diagram menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa).

Aplikasi diterapkan pada pengirim dan penerima, MMS dikirim dari pengirim ke penerima melalui jaringan MMS sehingga *deploy diagram* yang dihasilkan dapat dilihat pada Gambar diatas.

4.2.2. Analisa Penerapan Algoritma DCT Pada Steganografi

DCT (Discrete Cosine Transform) merupakan salah satu teknik transformasi yang mengubah suatu sinyal menjadi unsur komponen frekuensi dan sebaliknya komponen frekuensi di ubuh kembali kedalam suatu sinyal dengan menggunakan IDCT. Suatu gambar akan dikompresi yang dibagi ke dalam blok - blok 8x8, masing – masing blok akan di kenai DCT.

Pada metode DCT penyembunyian pesan terjadi apabila file yang dibutuhkan telah terinput. Pada saat itu, proses penyembunyian pesan telah siap dijalankan dengan cara merubah medianya menjadi matrik 8 x 8 dengan dengan dilanjutkan dengan proses kuantisasi dan *entropy coding*. Lalu didapatkanlah frekuansi yang bernilai tinggi dari kiri atas sampai kekanan bawah. Jadi yang dapat disisipkan dengan pesan adalah bagian frekuensi yang bernilai selain 0, -1 dan 1.

Coding suatu citra (single frame) :

1. Konversi color-space RGB ke YUV,
2. Partisi citra ke dalam blok 8x8-pixel,
3. Transformasi DCT 2-D untuk tiap blok,
4. Kuantisasi tiap koefisien DCT,
5. Kodekan dengan Runlength dan Huffman code untuk koefisien DCT terkuantisasi yang tidak nol.

Proses perhitungan koefisien DCT adalah untuk memperoleh nilai koefisien DCT (koefisien Ac dan dc), cara perhitungannya secara *cosines*.

4.2.3 Contoh Perhitungan Metode *Discrete Cosine Transform* (DCT)

a. Perhitungan Matriks *Transform*

Gambar dibagi menjadi blok, dan masing – masing blok memiliki 8 x 8 pixel.

$$\text{Original} = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

Data matriks original dikurangi dengan 128 karena algoritma DCT bekerja pada rentang -128 sampai 127 sesuai dengan ketentuan pengolahan citra bewarna.

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

Buat dan cari nilai untuk matriks DCT untuk matrik T dan buat matrik transposenya untuk matrik T^t.

T (i, j) = $Ci\ cos\frac{(2j+1)i\pi}{2N}$,Dimana $Ci = \sqrt{\frac{1}{N}}\ (i = 0), Ci = \sqrt{\frac{2}{N}}\ (i > 0)$

Maka dengan menggunakan rumusan matriks diatas dapat dihitung nilai matriks T mulai dari T (0, 0) sampai T (7, 7).

T (0, 0) = $\frac{1}{\sqrt{N}} = \frac{1}{\sqrt{8}} = 0.3536$

T (0, 1) = $\frac{1}{\sqrt{N}} = \frac{1}{\sqrt{8}} = 0.3536$

Begitu seterusnya hingga T (0, 7), lalu dengan T(1, 0) sampai T (7, 7) menggunakan persamaan satu lagi.

T (1, 0) = $\sqrt{\frac{2}{N}} \cos\frac{(2j+1)i\pi}{2N} = \sqrt{\frac{2}{8}} \cos\frac{(2.0+1)1.180°}{2.8} = 0.4904$

T (1, 1) = $\sqrt{\frac{2}{N}} \cos\frac{(2j+1)i\pi}{2N} = \sqrt{\frac{2}{8}} \cos\frac{(2.1+1)1.180°}{2.8} = 0.4157$

Maka dari perhitungan diatas didapatkan nilai untuk matriks T dan matriks transpose T adalah sebagai berikut.

$$T = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & -0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.0904 & -0.2778 & -0.2778 & 0.0904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & -0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & -0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

$$T^t = \begin{bmatrix} 0.3536 & 0.4904 & 0.4619 & 0.4157 & 0.3536 & 0.2778 & 0.1913 & 0.0975 \\ 0.3536 & 0.4157 & 0.1913 & -0.0975 & -0.3536 & -0.4904 & -0.4619 & -0.2778 \\ 0.3536 & 0.2778 & -0.1913 & -0.0904 & -0.3536 & 0.0975 & 0.4619 & 0.4157 \\ 0.3536 & 0.0975 & -0.4619 & -0.2778 & 0.3536 & 0.4157 & -0.1913 & -0.4904 \\ 0.3536 & -0.0975 & -0.4619 & -0.2778 & 0.3536 & -0.4157 & -0.1913 & 0.4904 \\ 0.3536 & -0.2778 & -0.1913 & 0.0904 & -0.3536 & -0.0975 & 0.4619 & -0.4157 \\ 0.3536 & -0.4157 & -0.1913 & 0.0975 & -0.3536 & -0.4904 & -0.4619 & 0.2778 \\ 0.3536 & -0.4904 & 0.4619 & -0.4157 & -0.3536 & -0.2778 & 0.1913 & -0.0975 \end{bmatrix}$$

Dengan menggunakan persamaan DCT, cari matriks D dimana matriks D akan digunakan untuk kuantisasi lanjut.

D = T . Z Dimana

$$Z = M \cdot T^{t}$$

Z(k,0) = 0.3536 (xk0 + xk1 + xk2 + xk3 + xk4 + xk5 + xk6 + xk7)

Z(k,1) = 0.4904(xk0 – xk7) + 0.4157(xk1 – xk6) + 0.2778(xk2 – xk5) + 0.0975(xk3 – xk4)

Z(k,2) = 0.4619(xk0 + xk7) + 0.1919(xk1 + xk6) – 0.1913(xk2 + xk5) – 0.4619(xk3 + xk4)

Z(k,3) = 0.4157(xk0 – xk7) – 0.0975(xk1 – xk6) - 0.4904(xk2 – xk5) – 0.2778(xk3 –xk4)

Z(k,4) = 0.3535(xk0 + xk7) – 0.3536(xk1 + xk6) – 0.3536(xk2 + xk5) + 0.3536(xk3 + xk4)

Z(k,5) = 0.2778(xk0 – xk7) – 0.4904(xk1 – xk6) + 0.0975(xk2 – xk5) +0.4157 (xk3 – xk4)

Z(k,6) = 0.1913(xk0 + xk7) – 0.4619(xk1 + xk6) + 0.4619(xk2 + xk5) – 0.1913(xk3 + xk4)

Z(k,7) = 0.0975(xk0 – xk7) – 0.2778(xk1 + xk6) + 0.4175(xk2 – xk5) – 0.4904 (xk3 – xk4)

Dimana k = 0, 1, 2, . . ., 7

$$Z = \begin{bmatrix} 1.4 & 8.83 & 20.32 & 7.48 & 22.63 & 5 & 8.42 & 1.76 \\ 67.9 & 53.5 & 2.24 & 36.62 & 1.41 & -0.55 & -13.16 & -11.71 \\ 111.7 & 92.88 & 22,39 & 54.37 & 28.99 & -11.2 & 2.22 & 4.69 \\ 107.1 & 72.25 & 1.14 & 57.38 & 41.37 & 1.35 & -0.08 & -16.1 \\ 70.7 & 25.55 & -6.88 & 22.63 & 26.17 & 2.54 & -5.56 & -6.6 \\ 65.4 & -56.11 & 2.28 & 28.19 & -26.17 & 10.14 & -15.85 & -0.18 \\ 28.6 & -10.54 & 9.33 & -1.42 & 15.2 & 4.3 & -17.25 & -0.54 \\ 6 & -23.84 & -0.04 & -2.67 & -20.15 & 0.34 & -14.32 & -2.94 \end{bmatrix}$$

D = T Z

D(0,k) = 0.3536 (z0k + z1k + z2k + z3k + z4k + z5k + z6k + z7k)

D(1,k) = 0.4904(z0k – z7k) + 0.4157(z1k – z6k) + 0.2778(z2k – z5k) + 0.0975(z3k – z4k)

D(2,k) = 0.4619(z0k + z7k) + 0.1919(z1k + z6k) – 0.1913(z2k + z5k) – 0.4619(z3k + z4k)

D(3,k) = 0.4157(z0k – z7k) – 0.0975(z1k – z6k) - 0.4904(z2k – z5k) – 0.2778(z3k –z4k)

D(4,k) = 0.3535(z0k + z7k) – 0.3536(z1k + z6k) – 0.3536(z2k + z5k) + 0.3536(z3k + z4k)

D(5,k) = 0.2778(z0k – z7k) – 0.4904(z1k – z6k) + 0.0975(z2k – z5k) +0.4157 (z3k – z4k)

D(6,k) = 0.1913(z0k + z7k) – 0.4619(z1k + z6k) + 0.4619(z2k + z5k) – 0.1913(z3k + z4k)

D(7,K) = 0.0975(z0k – z7k) – 0.2778(z1k + z6k) + 0.4175(z2k – z5k) – 0.4904 (z3k – z4k)

Matriks D sekarang berisi dengan koefisien DCT, dimana data yang terletak pada kiri atas merupakan korelasi dari frekuensi – frekuensi rendah dari data original. Sedangkan yang terletak pada kanan bawah merupakan korelasi dari frekuensi – frekuensi tinggi dari data original. Setelah itu lakukan proses kuantisasi dengan *Quality Level* 50.

$$D = \begin{bmatrix} 162.3 & 40.6 & 20 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6 & 11.5 & -6 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12 \\ -10 & 11.2 & 7.8 & -16.3 & 21.5 & 0 & 5.9 & 10.7 \end{bmatrix}$$

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Persamaan matriks kuantisasi adalah sebagai berikut, dimana round berarti mendekatkan nilai hasil pembagian ke pembulatan bilangan integer terdekat.

$$C_{ij} = \text{round}\ \frac{Dij}{Qij}$$

$C_{00} = \text{round}\ \frac{D00}{Q00} = \text{round}\ \frac{162.3}{16} = 10$

$C_{01} = \text{round}\ \frac{D01}{Q01} = \text{round}\ \frac{40.6}{11} = 4$, Dan begitu seterusnya sampai C_{77}

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4.2.4 Analisa Penerapan Algoritma RC6 dalam Steganografi DCT

Algoritma RC6 merupakan algoritma sederhana, fungsi yang digunakan merupakan fungsi yang sederhana dan hanya mengandalkan prinsip *iterated cipher* untuk keamanan.

Tampilan hasil enkripsi dan data hasil enkripsi yang diterima harus diperhatikan, hal ini dikarenakan pada data hasil enkripsi, setiap karakternya akan memiliki panjang 8 bit, sedangkan sebagian telepon selular hanya dapat menampilkan karakter dengan panjang 7 bit. Dengan demikian dalam penerapan algoritma RC6 pada SMS karakter-karakter yang akan dienkripsi diubah kedalam nilai ASCII, dimana nilai karakter dalam table ASCII ditambah table karakter special adalah 0 sampai dengan 255, artinya satu karakter ASCII akan diwakili oleh 8 bit, dimana $2^8 = 256$. Sehingga, dalam 1 blok plainteks (32 bit) akan menyimpan 4 karakter dan setiap kali iterasi, maka akan diambil 16 karakter dari plainteks.

Apabila panjang plainteks atau panjang kunci kurang dari 16 karakter, maka akan dilakukan *padding*, yaitu dengan menambah karakter “0” (nol) di akhir teks, sehingga panjang teks mencukupi 16 karakter.

Layar pada sebagian besar telepon selular hanya dapat menampilkan karakter dengan panjang 7 bit dan pesan yang teleh terenkripsi akan berbentuk *binary*, sehingga layar tidak akan menampilkan dengan semestinya.

Algoritma RC6 yang akan digunakan dalam aplikasi dibangun dengan w sebesar 32 bit, r sebesar 20 kali putaran dan panjang kunci beragam lebih dari 1 karakter (8 bit). Langkah-langkah algoritma RC6 dalam pelaksanaan tugas akhir ini akan dikelompokkan kedalam beberapa bagian, yaitu .

1. Pembangkit Sub Kunci

   Kunci dari pengguna ini akan dimasukkan oleh pengguna pada saat akan melakukan proses enkripsi dan dekripsi. Kunci ini memiliki tipe data *string* dan memiliki panjang 16 *byte* (16 karakter)
2. Baca masukkan untuk proses enkripsi

   Yang dilakukan pada tahapan ini adalah membaca teks yang menjadi masukan pada proses enkripsi, yaitu *field* dari aplikasi steganografi, *field* nya adalah isi pesan.
3. Enkripsi meliputi *whitening* awal, iterasi, dan *whitening* akhir.
4. Baca masukkan untuk Proses Dekripsi

   Yang dilakukan pada tahapan ini adalah membaca teks yang telah disisipi yang menjadi masukkan pada proses dekripsi, yaitu *record* dari hasil pesan yang telah disisipi pada pengirim dan menjadi *field* pesan pada penerima.
5. Dekripsi merupakan kebalikan dari proses enkripsi

Langkah-langkah diatas akan dijelaskan dalam algoritma-algoritma berikut :

A. Algoritma Pembangkit Sub Kunci

```
Kamus
      Type Word32 : 32 bit {tipe data 32 bit}
      Kunci : string {kunci yang dimasukkan oleh pengguna}
      I, j, c, s, v : integer
      A : integer
      B : integer
      S : array [0..43] of word 32
      L : array [0..43] of word 32
Function
```

```
      ROTL (X:Word32; Y:integer) → Word 32 {fungsi untuk merotasi
      bit sebanyak variable kedua}
Algoritma
      Input (Kunci)
      S[0] ← b7e15163
      For i ← 1 to 43 do
            S[i] ← S[i-1] + 9e3779b9
      Endfor
      A ← B ← i ← j ← 0
      V ← 44
      If (c > v) then
            v ← c
            v ← v*3
      For s ← 1 to v do
            A ← S[i] ← ROTL ((S[i] + A + B). 3)
            B ← L[j] ← ROTL (L[j] + A + B, A + B)
            i ← (i+1) mod 44
            j ← (j+1) mod c
      Endfor
```

Algoritma 4.1 Pembangkit sub kunci

B. Algoritma Baca File Masukan Proses Enkripsi

```
Prosedur Baca_Masukan_Proses_Enkripsi
      {Input : Field masukan belum dibaca}
      {Output : Field masukan dibaca per 16 karakter dan ditampung
      dalam buffer. Pada proses isi pesan, field nya dalah isi
      pesan}
Kamus
      Field_masukan : string
      Buff : array [0..15] of char
      i : integer
Algoritma
      Input (field_masukan)
      i ← 0
      while (I <= 15) and not (EOF) do
            Read (field_masukan, Buff[i])
```

```
    Endwhile
```

Algoritma 4.2 Baca Masukan untuk Proses Enkripsi

C. Algoritma *Whitening* Awal

```
Prosedur  Whitening_awal
    {input : blok kedua dan keempat belum dijumlahkan dengan sub
    kunci}
    Output : blok kedua dan keempat yang telah dijumlahkan
    dengan sub kunci}

Kamus
    Type word32 : 32 bit (tipe data sebesar 32 bit)
    X : word32 array [0..3] (blok enkripsi/plainteks)
    S : Array [0..43] of word 32 (sub kunci)

Algoritma
    X[1] ← X[1] + S[0]
    X[3] ← X[3] + S[1]
```

Algoritma 4.3 *Whitenig* Awal

D. Algoritma Iterasi

```
Prosedur Iterasi
    {input : keempat blok setelah whitening awal belum diproses}
    output : keempat blok yang telah diproses dan saling
    dipertukarkan}

Kamus
    Type word32 : 32 bit {tipe data sebesar 32 bit}
    X : word array [0..3] {blok enkripsi/plainteks}

Function
    ROTL(X : Word32; Y : integer) → word32
    {merotasi bit kekiri sebanyak variable kedua}
    Temp : word32
    u, t : word32
    I : integer

Algoritam
    For i ← 1 to 20 do
        t ← ROTL ((X[1]*(2*X[1]+1)), 5)
```

```
            u ← ROTL ((X[3]*(2*X[3]+1)), 5)
            X[0] ← (ROTL((X[0] XOR t), u)) + S[2*i]
            X[2] ← (ROTL((X[2] XOR u), t)) + S[2*I + 1]
            Temp ← X[0]
            X[0] ← X[1]
            X[1] ← X[2]
            X[2] ← X[3]
            X[3] ← Temp
      End for
```

Algoritma 4.4 Iterasi

E. Algoritma *Whitening* Akhir

```
Prosedur Whitenig_akhir
      {input : blok pertama dan ketiga belum dijumlahkan dengan
      sub kunci}
      Output : blok pertama dan ketiga yang telah dijumlahkan
      dengan sub kunci}

Kamus
      Type word32 : 32 bit (tipe data sebesar 32 bit)
      X : word32 array [0..3] (blok enkripsi/plainteks)
      S : Array [0..43] of word 32 (sub kunci)

Algoritma
      X[0] ← X[0] + S[42]
      X[2] ← X[0] + S[43]
```

Algoritma 4.5 *Whitening* Akhir

F. Algoritma Baca File Masukan Proses Dekripsi

```
Prosedur Baca_File_Masukan_Proses_Dekripsi
      {input : Field masukan berupa chiperteks}
      {Output : Field pada isi pesan yang berupa chiperteks dibaca
      per 16 karakter dan ditampung dalam buffer}

Kamus
      Field_masukan : string
      Buff : array [0..15]
      i : integer
```

```
Algoritma
      Input (field_masukan)
      i ← 0
      while (i <= 15 ) and not (field_masukan.EOF) do
            Read (isi_kolom, Buff[i])
      Endwhile
```

Algoritma 4.6 Baca File Masukan untuk Proses Dekripsi

G. Algoritma Dekripsi

```
Prosedur Dekripsi
      {input : keempat blok belum diproses
      Output : keempat blok yang telah diproses dan saling
      dipetukarkan}

Kamus
      Type word32 : 32 bit {tipe data sebesar 32 bit}
      X : word32 array [0..3] {blok dekripsi/ciperteks}

Function
      ROTL (X:word32; Y:integer) → word32 {merotasi bit kekiri
      sebanyak variable kedua}
      Temp : word32
      U, t : word32
      I : integer

Algoritma
      X[2] ← X[2] - S[43]
      X[0] ← X[0] – S[42]
      For i ← 20 downto 1 do
            Temp ← X[3]
            X[3] ← X[2]
            X[2] ← X[1]
            X[1] ← X[0]
            X[0] ← Temp
            u ← ROTL ((X[3]*(2*X[3]+1)), 5)
            t ← ROTL ((X[1]*(2*X[1]+1)), 5)
            X[2] ← (ROTR(X[2] – S[2*i+1]), t) XOR u)
            X[0] ← (ROTR(X[0] – S[2*i]), u) XOR t)
```

```
    End for
    X[3] ← X[3] – S[1]
    X[1] ← X[1] – S[0]
```

Algoritma 4.7 Dekripsi

4.2.5 Perhitungan Manual Algoritma RC6

Pada perhitungan manual algoritma RC6 ini diberikan kunci sebesar 16 byte dan plainteks sebesar 128 bit (16 *byte*). Kunci dan plainteks yang menjadi contoh masing-masing sebagai berikut :

Kunci : `endrikomartafori`

Plainteks : `teknik informasi`

Langkah pertama adalah membagi plainteks kedalam 4 blok yaitu A, B, C, D, yang masing-masing blok yang terdiri dari 32 bit (4 karakter)

```
 A    B     C     D
Tekn ik i  nfor  masi
```

Ubah tiap karakter dalam masing-masing blok kedalam nilai ASCII, selanjutnya ubah nilai ASCII tersebut menjadi bilangan biner masing-masing sepanjang 18 bit, sehingga pada masing-masing blok akan dihasilkan bilangan biner sepanjang 32 bit.

Blok A

Plainteks	t	e	k	n
ASCII	116	101	107	110
Biner	01110100	01100101	01101011	01101110

Blok B

Plainteks	i	k		i
ASCII	105	107	160	105
Biner	01101001	01101011	00100000	01101001

Blok C

Plainteks	n	f	o	r
ASCII	110	102	111	114
Biner	0110110	01100110	01101111	01110010

Blok D

Plainteks	m	a	s	i

```
ASCII              109         97          115         105
Biner           01101101    01100001    01110011    01101001
```

Kemudian bilangan biner digabungkan kembali, dengan aturan byte pertama plainteks diletakkan pada lest significan bit blok A. Dan byte terakhir plainteks diletakkan pada most significan bit blok D

```
Blok A          : 01101110011010110110010101110100
Dalam desimal     1.852.532.084

Blok B          : 01101001001000000110101101101001
Dalam desimal     1.763.732.329

Blok C          : 01110010011011110110011001101110
Dalam desimal     1.919.903.342

Blok D          : 01101001011100110110000101101101
Dalam desimal     1.769.169.261
```

Setelah didapat nilai pada masing-masing blok, maka dilanjutkan dengan langkah-langkah berikut (Perhitungan manual pembangkit sub kunci dapat dilihat pada Lampiran A) :

1. Whitening Awal

Whitening awal, dengan menjumlahkan B dengan sub kunci S(0), dan D dengan sub kunci S(1). Penjunlahan dilakukan dalam modulo 2^{32}

```
B     = B + S(0)
D     = D + S(1)
B     = 1.763.732.329 + 4.033.202.597 mod 2^32
      = 5.796.934.926 mod 4.294.967.296
      = 1.501.967.630
D     = 1.769.169.261 + 1.623.197.347 mod 2^32
      = 3.392.366.608 mod 4.294.967.296
      = 3.392.366.608
```

2. Iterasi

Iterasi dilakukan sebanyak 20 kali. Setiap iterasi mengikuti aturan sebagai berikut :

```
t ← ROTL ((X[1]*(2*X[1]+1)), 5)
u ← ROTL ((X[3]*(2*X[3]+1)), 5)
```

```
X[0] ← (ROTL ((X[0] XOR t), u)) + S[2*i]
X[2] ← (ROTL ((X[2] XOR u), t)) + S[2*i + 1]
Temp ← X[0]
X[0] ← X[1]
X[1] ← X[2]
X[2] ← X[3]
X[3] ← Temp
```

Nilai t dan u didapat dari blok B dan D diproses dengan fungsi f(x) = x(2x+1), kemudian dilanjutkan dengan menggeser nilai t dan u ke kiri sejauh 5 bit.

```
t = (B * (2*B+1))
  = (1.501.967.630 * (2 * 1.501.967.630 + 1)) mod 2^32
  = (1.501.967.630 * 3.003.935.261) mod 4.294.967.296
  = 4.511.813.524.637.601.430 mod 4.294.967.296
  = 4.241.739.414
t : (dalam biner)       11111100110100111100111010010110
t : (digeser 5 bit)     10011010011110011101001011011111
t : (dalam desimal)     2.591.675.103
```

Nilai 5 bit terakhir dari t yaitu 11111, atau dalam desimal sebesar 31, akan dipergunakan sebagai nilai penggeser blok C pada proses berikutnya, sejauh 31 bit

```
u = (D * (2 * D + 1) mod 2^32
  = (3.392.366.608 * (2 * 3.392.366.608 + 1)) mod 2^32
  = (3.392.366.608 * (6.784.733.216 + 1)) mod 2^32
  = (3.392.366.608 * 6.784.733.217) mod 2^32
  = 4.569.558.335.829.666.320 mod 4.294.967.296
  = 2.341.300.752
u : (dalam biner)       10001011100011010110101000010000
u : (digeser 5 bit)     01110001101011010100001000010001
u : (dalam desimal)     1.907.180.049
```

Nilai 5 bit terakhir dari u yaitu 10001, atau dalam desimal sebesar 17, akan dipergunakan sebagai pergeseran blok A pada proses berikutnya, sejauh 17 bit.

Maka didapatkan nilai-nilai sebagai berikut :

```
t = 2.591.675.103
u = 1.907.180.049
penggeser t = 31
penggeser u = 17
```

Langkah selanjutnya adalah memproses blok A dan C dengan nilai-nilai yang telah dihasilkan.

```
A = (ROTL ((A XOR t), u)) + S[2*i]
A : 1.852.532.084, dalam biner 01101110011010110110010101110100
t : 2.591.675.103, dalam biner 10011010011110011101001011011111 ⊕
A : (hasil xor)                11110100000100101011011110101011
A : (digeser 31 bit)           11111010000010010101101111010101
A : (dalam desimal)            4.194.917.333
```

Niai A dijumlahkan dengan sub kunci S(2), dalam modulo 2^32 :

```
A = 4.194.917.333 + 1.929.374.348 mod 2^32
  = 6.124.291.681 mod 4.294.967.296
  = 1.829.324.385
C = (ROTL((C XOR u), t)) + S[2*i+1]
C : 1.919.903.342, dalam biner 01110010011011110110011001101110
u : 1.907.180.049, dalam biner 01110001101011010100001000010001 ⊕
C : (hasil xor)                00000011110000100010010001111111
C : (digeser 16 bit)           00100100011111110000001111000010
C : (dalam desimal)            612.303.810
```

Nilai C dijumlahkan dengan sub kunci S(3), dalam modulo 2^32

```
C = 612.303.810 + 4.270.616.488 mod 2^32
  = 4.882.920.298 mod 4.294.967.296
  = 587.953.002
```

Maka didapat nilai masing-masing blok adalah :

```
A     : 1.829.324.385
B     : 1.501.967.630
C     : 587.953.002
D     : 3.392.366.608
```

Langkah berikutnya adalah mempertukarkan nilai blok dengan aturan (A, B, C, D) ← (B, C, D, A), sehingga pada iterasi pertama, didapat nilai pada masing-masing blok sebagai berikut :

```
A      : 1.501.967.630
B      : 587.953.002
C      : 3.392.366.608
D      : 1.829.324.385
```

Nilai masing-masing blok akan dilanjutkan pada iterasi berikutnya sebanyak 20 kali.

4.2.6 Proses Penyisipan Pesan Dengan Least Significant Bit (LSB) 1 Bit

Pada sebuah citra JPEG 8x8 piksel yang telah dikompresi dengan menggunakan algoritma DCT akan disisipkan pesan yang berbunyi “ME”.

Kode ASCII dari pesan tersebut sebagai berikut:

77 69

Kode ASCII tersebut untuk selanjutnya diubah menjadi 7 bit kode-kode biner sehingga di dapat:

1001101 1000101

Matrik hasil kompresi dengan algoritma DCT adalah sebagai sebagai berikut;

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Karena proses entropy coding yang digunakan untuk menyimpan data dipengaruhi oleh jumlah 0 pada matrik, oleh karena itu angka -1, 0, dan 1 pada matriks tidak diikutsertakan dalam penyisipan.

Blok pixel yang bisa disisipi dari matrik diatas adalah sebagai berikut:

10 4 2 5

3 9 2

-7 -5 -2

-3 -5

-2 2

Kode binernya seperti dibawah ini :

00000000 00001010 00000000 00000100 00000000 00000010 000000 00000101

00000000 00000011 00000000 00001001 00000000 00000010

11111111 00000111 11111111 00000101 11111111 00000010

11111111 00000011 11111111 00000101

11111111 00000010 00000000 00000010

Untuk selanjutnya, tiap bit kode biner pesan digunakan untuk menggantikan bit terakhir dari kode biner yang dapat disisipi pesan. Proses penggantian dilakukan terurut, menurut baris ataupun kolom. Pada percobaan ini digunakan kolom.

Angka yang dicetak tebal pada bilangan biner dibawah ini berarti telah mengalami perubahan nilai setelah proses penggantian dilakukan, maka kode biner matrik citra tersebut menjadi:

00000000 00001010**1** 00000000 00000100 00000000 00000010 000000 00000101

00000000 00000010**0** 00000000 00001001 00000000 00000010**1**

11111111 00000110**0** 11111111 00000101 11111111 00000010

11111111 00000011 11111111 00000100**0**

11111111 00000010**1** 00000000 00000010

Matrik biner tersebut dikembalikan lagi menjadi decimal sehingga di dapat nilai matriknya adalah:

$$C = \begin{bmatrix} \underline{\mathbf{11}} & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ \underline{\mathbf{2}} & 9 & 1 & \underline{\mathbf{3}} & 1 & 0 & 0 & 0 \\ \underline{\mathbf{-6}} & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & \underline{\mathbf{-4}} & 0 & -1 & 0 & 0 & 0 & 0 \\ \mathbf{-3} & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrik ini akan dipetakan kembali ke bentuk citra. Ekstraksi pesan dapat dengan mudah dilakukan dengan mengambil bit terakhir dari kode biner citra.

Jika diperhatikan, penggantian bit terakhir tersebut tidak terlalu berpengaruh terhadap nilai citra. Ada tiga kemungkinan yang terjadi setelah penggantian bit terakhir, yakni:

1. Nilainya citranya tetap,
2. Nilai citranya berkurang 1,
3. Nilai citranya bertambah 1.

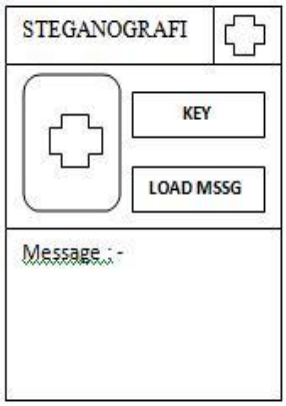
Penyisipan pesan pada matrik frekuensi memang akan mempengaruhi area tersebut, namun karena penyisipan data dilakukan dalam domain frekuensi, perubahan yang terjadi tidak mengubah bagian yang tampak pada gambar.

4.2.7 Perancangan Interface

Berikut adalah contoh tampilan interface aplikasi steganografi DCT dan kriptografi RC6 yang berjalan pada sistem operasi android.

A. Halaman Utama

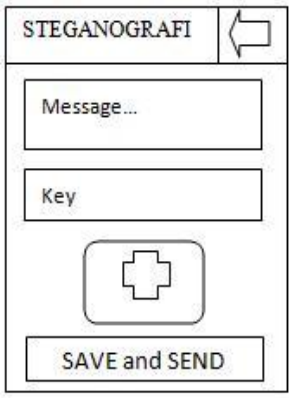
Halaman utama pada aplikasi ini akan tampil, jika *shortcut* aplikasi steganografi DCT dan kriptografi RC6 di klik pada layar menu utama *handphone*, tampilan utama aplikasi ini juga berfungsi untuk mengekstrasi gambar yang sudah disisipi pesan. Tampilan halaman utama pada aplikasi dapat dilihat seperti gambar berikut :



Gambar 4.12 Tampilan Utama Stegano DCT dan Kripto RC6

B. Tulis Pesan

Menu tulis pesan ini tampil, apabila tombol tulis pesan pada aplikasi steganografi DCT dan kriptografi RC6 di klik. Pada interface tulis pesan, user menginputkan pesan yang akan disisipi serta menginputkan kunci enkripsi, kemudian menginputkan gambar yang dijadikan media penyisipan. Setelah itu, user mengklik tombol kirim, kemudian pesan gambar yang sudah berisi pesan terenkripsi dikirim melalui media yang kita inginkan.



Gambar 4.13 Tampilan *New Message*

BAB V IMPLEMENTASI DAN

PENGUJIAN

5.1 Fase *Construction*

Pada bab ini memasuki fase *construction* model RUP yang berisikan penjelasan mengenai implementasi perangkat lunak yang meliputi implementasi perangkat lunak, batasan implementasi, implementasi kelas, dan implementasi antar muka.

5.1.1 Implementasi Perangkat Lunak

Implementasi aplikasi ini dibangun dengan menggunakan algoritma enkripsi *Rivest Code 6* dan algoritma steganografi DCT yang dibangun dengan menggunakan bahasa pemograman Java, dan XML.

Pemilihan perangkat lunak ini didasarkan pertimbangan sebagai berikut :

1. Java merupakan bahasa pemograman berorientasi object yang banyak diterapkan dalam aplikasi mobile.
2. Untuk pemograman di sistem Operasi Android menggunakan bahasa pemograman Java dan runtimenya menggunakan *Dalvik Virtual Machine*.
3. Untuk tampilan interface di Android menggunakan pemograman XML.

5.1.2 Batasan Implementasi

Pada tugas akhir ini, perangkat lunak yang dibangun memiliki batasan berikut:

1. Perangkat lunak tidak dapat melakukan akses ke *memory* dalam kartu SIM.
2. Perangkat lunak yang dibangun dapat dijalankan pada telepon selular bersistem operasi Android dengan spesifikasi minimal versi 2.2 dan menggunakan kartu GSM.

5.1.3 Lingkungan Implementasi

Implementasi yang dilakukan menggunakan sebuah perangkat komputer untuk membangun perangkat lunak dan sebuah telopon selular bersistem operasi Android yang digunakan untuk melakukan uji perangkat lunak yang telah dibangun.

Perangkat komputer yang digunakan untuk melakukan implementasi secara virtual memiliki spesifikasi sebagai berikut :

1. Prosessor Intel Core i5 2.4 GHz
2. RAM 2 GB
3. *Hard Disk* 320 GB
4. Perangkat masukan *Keyboard* dan *Mouse*
5. Perangkat keluaran monitor

Adapun perangkat lunak yang digunakan dalam melakukan implementasi adalah sebagai berikut :

1. Sistem Operasi Windwos 7 Ultimate
2. *Java Development Kit*
3. Eclips Juno
4. *Plugins* Eclips Android Development Tools 1.6
5. Android SDK 2.2 dan *Android Virtual Device*

5.1.4 Implementasi Kelas

Kelas-kelas yang telah dirancang diimplementasikan dengan menggunkan bahasa pemograman java.

A. Deskripsi Kelas

Pada table 5.1 dapat dilihat daftar implementasi kelas-kelas yang ada pada perangkat lunak beserta keterangannya.

Tabel 5.1 Implementasi Kelas

Nama Kelas	Nama File	Keterangan
Penerima	LoadMessage.java	Kelas ini merupakan tampilan utama dari aplikasi steganografi, ketika aplikasi dijalankan, maka kelas ini yang pertama dipanggil
Pengirim	NewMessage.java	Kelas ini merupakan tampilan untuk melakukan penulisan, pengenkripsian, dan pengririman pesan
Penyisipan	Kalkulator.java	Kelas ini merupakan kelas untuk menyisipkan pesan kedalam gambar
RC6	RC6.java	Kelas ini merupakan kelas algoritma enkripsi dekripsi dari RC6, serta algoritma dari penjadwalan kunci RC6

B. Operasi dan Atribut

1. Kelas Penerima

   Daftar hasil implementasi operasi dai kelas Interface dapat dilihat pada table 5.2.

Tabel 5.2 Daftar Implementasi Operasi Kelas Penerima

Nama Operasi	Visibility (*private, public)*	Implementasi
K	*Public*	`public void IFL()`
IBL	*Public*	`public void IFL()`
BTG	*Public*	`public void IFL()`
TVM	*Public*	`public void IFL()`
ETP	*Public*	`public void IFL()`

2. Kelas Pengirim

   Daftar implementasi operasi dari kelas Pengirim dapat dilihat pada Tabel 5.3 dan atribut yang digunakan dapat dilihat pada table 5.4

Tabel 5.3 Daftar Implementasi Operasi Kelas Pengirim

Nama Operasi	Visibility (*private, public)*	Implementasi
K	*Public*	`public void IFN()`
IBA	*Public*	`public void IFN()`
BTS	*Public*	`public void IFN()`
ETN	*Public*	`public void IFN()`
ETM	*Public*	`public void IFN()`
ETK	*Public*	`public void IFN()`

Tabel 5.4 Daftar Implementasi Atribut Kelas *NewMessage*

Nama Operasi	Visibility (*private, public)*	Implementasi
STG	Public	`public void STG()`

2. Kelas Penyisipan

Table 5.5 Daftar Implementasi Operasi Kelas Penyisipan

Nama Operasi	Visibility (*private, public)*	Implementasi
AM	*Public*	`public Bitmap AM(Bitmap Data, String Pesan, String Kunci)`
EK	*Private*	`private int[] EK(String Pesan, String Kunci)`
GM	*Public*	`public String GM(Bitmap Data, String Kunci)`
DK	*Private*	`private String DK(String temp, String Kunci)`
ITB	*Private*	`private String ITB(int data)`

3. RC 6

   Daftar implementasi operasi dari kelas RC 6 dapat dilihat pada table 5.6 dan atribut yang digunakan dapat dilihat pada table 5.7

Table 5.6 Daftar Implemantasi Operasi Kelas RC 6

Nama Operasi	Visibility (*private, public*)	Implementasi
ConvByteWord	*Private*	`private int[] convBytesWords(byte[] key, int u, int c)`
GenerateSubKey	*Private*	`private int[] generateSubkeys(byte[] key)`
Rotl	*Private*	`private int rotl(int val, int pas)`
Rotr	*Private*	`private int rotr(int val, int pas)`
DecryptBloc	*Private*	`private byte[] decryptBloc(byte[] input)`
EncryptBloc	*Private*	`private byte[] encryptBloc(byte[] input)`
PaddingKey	*Private*	`private static byte[] paddingKey(byte[] key)`
Encrypt	*Public*	`public byte[] encrypt(byte[] data, byte[] key)`
Decrypt	*Public*	`public byte[] decrypt(byte[] data, byte[] key)`
DeletePadding	*Private*	`private byte[] deletePadding(byte[] input)`

Tabel 5.7 Daftar Implementasi Atribut Kelas RC 6

Nama Atribut	Visibility (*public, private*)	Implementasi

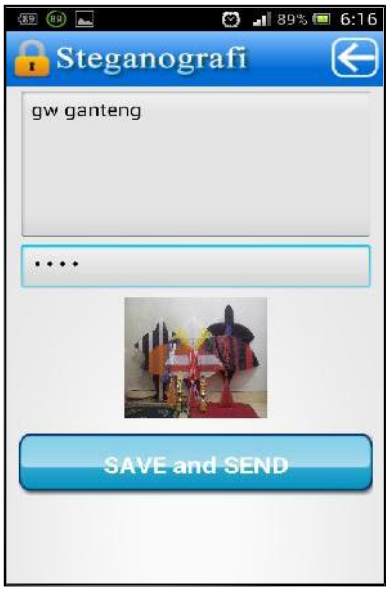
w	*Private*	`private int w=32`
r	*Private*	`private int r=20`
Pw	*Private*	`private int Pw=0xb7e15163`
Qw	*Private*	`private int Qw=0x9e3779b9`
S	*Private*	`private int[] S`

5.1.5 Implementasi Antar Muka

Sub bab ini berisi gambar-gambar hasil implementasi antar muka dari perangkat lunak yang telah dibangun. Gambar hasil implementasi tersebut merupakan gambar dari Print Screen pada handphone.

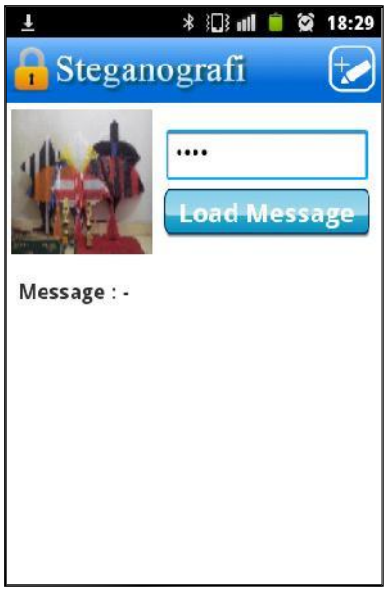
Gambar 5.1 Tampilan Awal Aplikasi

Gambar 5.2 Tampilan Untuk Menyisipkan Pesan

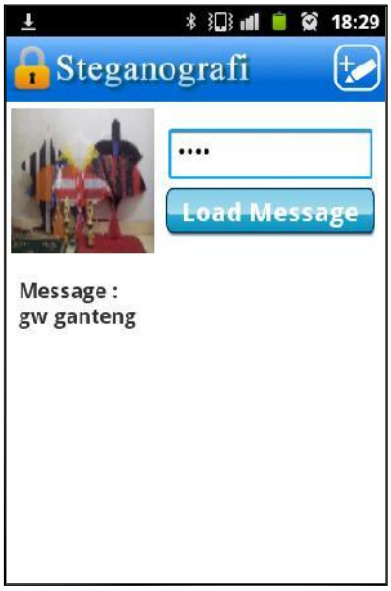


Gambar 5.3 Tampilan Hasil Penyisipan

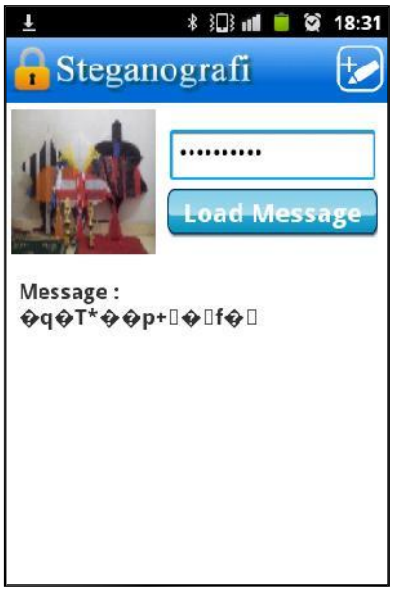
Gambar 5.4 Tampilan Media Pengiriman



Gambar 5.5 Tampilan Penerima Pesan



Gambar 5.6 Tampilan Aplikasi Dienkripsi Dengan Kunci Yang Benar



Gambar 5.7 Tampilan Aplikasi Dienkripsi Dengan Kunci Yang Salah

5.2 Fase *Transition*

Alur terakhir dari RUP yaitu *transition* dimana dalam fase ini dilakukan *deployment* aplikasi serta melakukan testing pengujian dari aplikasi, dari pengujian tersebut dapat diambil kesimpulan terhadap aplikasi yang dibuat.

5.2.1 Pengujian

Pengujian yang dilakukan dibagi menjadi empat yaitu pengujian secara *blackbox*, pengujian panjang pesan dan *handphone*, serta pengujian keamanan pesan terenkripsi.

5.2.1.1 Pengujian *Blackbox*

Pengujian dengan menggunakan metode *blackbox* ini dilakukan dengan mengevaluasi cara kerja aplikasi yang telah dibuat apakah sesuai harap atau tidak.

Tabel 5.8 Pengujian *Blackbox*

No	Komponen Pengujian	Hasil yang diharapkan	Hasil Pengujian	Keterangan
1	Halaman Utama	Pengguna mengklik aplikasi Steganografi dan menampilkan halaman utama	Halaman utama dari aplikasi berhasil ditampilkan	Benar
2	*Button New Message* pada menu *Load Message*	Aplikasi berhasil menuju halaman *New Message*	Halaman *New Message* dari aplikasi berhasil ditampilkan	Benar
3	*Button Gallery*	Menampilkan *list* gambar pada *gelery handphone*	Gambar yang dipilih berhasil di *input* ke dalam aplikasi	Benar
4	*Button Load Message*	Menampilkan hasil dari steganografi dan enkripsi pada gambar	Teks yang telah disipkan berhasil ditampilkan jika kunci yang digunakan benar dan sebaliknya	Benar
5	*Button Save and Send*	Menampilkan media pengiriman aplikasi	Media *Bluetooth* , *Messaging*, dan lain-lain berhasil digunakan	Benar

5.2.1.2 Pengujian Panjang Pesan dan Handphone

A. Enkripsi dan Dekripsi Pesan Pada Pengirim dan Penerima

Pada bagian ini akan diberikan hasil pengujian dari proses enkripsi dan dekripsi dengan kasus-kasus tertentu. Kasus-kasus yang akan diujikan adalah sebagai berikut :

1. Menguji kebenaran enkripsi dan dekripsi,
2. Menguji panjang pesan hasil enkripsi dengan penggunaan kunci panjang dan penggunaan pesan panjang yang melebihi 160 karakter.
3. Menguji validasi format.
4. Menguji proses penyisipan pesan ke dalam gambar.

Hasil pengujian kasus-kasus tersebut dapat dilihat pada tabel **Lampiran B**.

B. Enkripsi dan Dekripsi pada *Merk Handphone* dan Versi Android Berbeda

Pengujian pengiriman pesan ini bertujuan untuk melakukan pengecekan terhadap pesan yang dikirimkan. Tata cara dari pengujian ini adalah melakukan pengiriman terhadap telepon selular bersistem operasi android versi 2.2 dan versi diatasnya dengan berbagai *merk* handphone.

Tabel 5.9 Pengujian pada Versi Android Berbeda

Merk dan Jenis Versi Android Penerima	**Pesan Yang dikirim**	**Pesan yang diterima**	**Fungsi Dekripsi**
Samsung Ver. 4.2.2 jelly bean	ini adalah pesan pertama dalam pengujian panjang pesan pada aplikasi steganografi dengan metode dct dan kriptografi rc6 di android	ini adalah pesan pertama dalam pengujian panjang pesan pada aplikasi steganografi dengan metode dct dan kriptografi rc6 di android	Berhasil
Sony Ericson Versi 4.1.2 jelly bean	ini adalah pesan pertama dalam pengujian panjang pesan pada aplikasi	ini adalah pesan pertama dalam pengujian panjang pesan pada	Berhasil

	steganografi dengan metode dct dan kriptografi rc6 di android	aplikasi steganografi dengan metode dct dan kriptografi rc6 di android	

5.2.1.3 Pengujian Keamanan Pesan Terenkripsi

Pengujian keamanan pesan terenkripsi dilakukan untuk mengetahui keamanan data yang telah disimpan dalam bentuk chiperteks terhadap serangan dari penyerang. Pengujian ini menggunkan jenis serangan *exhausitive attack* atau *brute force attack.* Percobaan yang dibuat untuk mengungkap plainteks atau kunci dengan mencoba semua kemungkinan kunci (*trial and error*)

Batasan pengujian keamanan data terenkripsi yaitu, pengujian dalam proses *attack* terhadap data terenkripsi dilakukan sebanyak 6 kali percobaan.

Asumsi yang dipergunakan dalam pengujian *attack*:

1. Chiperteks yang diuji adalah uraian-uraian dari pesan yang diterima
2. Kriptanalis memasukkan kemungkinan kunci yang digunakan secara acak.
3. Kunci yang dimasukkan sebesar 32 *byte* (16 karakter) sesuai besar panjang kunci plainteks dengan mempertimbangkan *case sensitive* nya.

Table 5.10 Pengujian Keamanan Data Terenkripsi Terhadap Serangan

No	Kunci Awal	Kunci Uji Coba	Hasil
1	endrikomar	1234567890	Tidak berhasil
2	endrikomar	abcdefghij	Tidak berhasil
3	endrikomar	12345abcde	Tidak berhasil
4	endrikomar	Abcde12345	Tidak berhasil
5	endrikomar	ENDRIKOmar	Tidak berhasil
6	endrikomar	endrikoMAR	Tidak berhasil

5.2.2 Kesimpulan Pengujian

Hasil pengujian aplikasi secara *blackbox* dapat dievaluasi bahwa hasil aplikasi yang dibuat sudah sesuai harapan.

Melalui hasil pengujian yang dilakukan pada telepon selular Android berbeda versi dan *merk*, pesan dapat disampaikan dengan baik dan fungsi dekripsi pesan berhasil dilakukan, dapat diketahui implementasi algoritma DCT dan RC6 untuk komunikasi melalui media MMS pada sistem operasi Android dapat direalisasikan dengan baik. Output gambar yang dihasilkan oleh penerima mengalami perubahan ukuran dan kualitas gambar,serta format gambar akan berubah menjadi PNG.

Pengujian keamanan hasil enkripsi menggunakan metode *exhausitive attack* atau *brute force attack* menunjukkan hasil bahwa dari 6 kali usaha percobaan kunci dengan kunci yang salah yang menerapkan fungsi *case sensitive*, persentase kegagalan sebesar 100%, artinya pesan tidak dapat dibuka dengan menggunkan kunci yang berbeda dengan kunci saat melakukan enkripsi.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisis, perancangan dan implementasi pada sistem yang telah dibuat dapat diambil kesimpulan sebagai berikut:

1. Telah berhasil dikembangkan perangkat lunak yang dapat melakukan steganografi pada citra JPEG. Kebutuhan fungsional dari perangkat lunak, seperti proses penyisipan dan ekstraksi pesan, serta penggunaan kunci sudah dapat dilakukan dengan benar. Pengembangan pada perangkat *mobile phone* juga telah berhasil dilakukan.
2. Panjang kunci yang digunakan dalam melakukan enkripsi minimal 1 karakter dan maksimal 10 karakter.
3. Kualitas citra yang dihasilkan bergantung dari besarnya ukuran pesan. Semakin besar ukuran pesannya maka kualitas gambar akan secara signifikan menurun.
4. Algoritma dapat melakukan pengamanan terhadap pesan yang dikirim melalui media MMS, dimana dengan percobaan serangan dengan menggunakan metode *exhaustive attack* terhadap kunci sebanyak 6 kali percobaan, pesan tidak dapat ditembus dengan menggunakan kunci yang salah.
5. Pengujian pada aplikasi ini telah memenuhi empat aspek keamanan data dan informasi yang meliputi *privacy*, *integrity*, *authentication*, dan *availability*. Adapun keempat aspek keamanan tersebut telah di uji pada pengujian keamanan data terenkripsi terhadap serangan, pengujian ekstrasi dengan kunci yang berbeda, pengujian penyisipan pesan, pengujian ekstrasi pesan, dan lain – lainnya.

6.2 Saran

1. Menambahkan dukungan terhadap format lain pada citra dan video yang mendukung kompresi DCT.
2. Menambahkan dukungan terhadap penyisipan pesan dengan menggunakan file dan citra
3. Mengganti algoritma kriptografi RC6 yang bersifat simetri ini dengan algoritma kriptografi asimetri, karena kunci rahasia pada algoritma kunci simetri untuk enkripsi digunakan juga untuk dekripsi. Apabila pertukaran kunci dilakukan melalui jalur MMS maka pertukaran kunci menjadi tidak aman.
4. Meneliti kembali penyebab maksimal 10 karakter kunci pada algoritma RC6 yang digunakan pada laporan ini, karena standard algoritma RC6 bisa menampung maksimal kunci sebanyak 16 karakter.

DAFTAR PUSTAKA

Android Developer Guide : http://developer.android.com , Diakses 15 Mei 2012

Darwiyanti, Sri dan Romi Satria Wahono. Pengenalan Unified Modeling Language (UML) .[Online] Available http://ilmukoputer.org/2006/08/05/pengantar-uml/, Diakases 15 Mei 2012 .

Gunawan Hariyanto, Paul, *Studi dan Implementasi Steganografi pada Video Digital di Mobile Phone dengan DCT Modification*, Laporan Tugas Akhir Sarjana, Program Studi Teknik Informatika Sekolah Teknik Elektro Dan Informatika Institut Teknologi, Bandung, 2008.

Marzuki, Ismail, *Rancang Bangun Aplikasi Untuk Penyisipan Text Dan File Ke Dalam Image Dan Audio File Dengan Metode Least Significant Bit (LSB)*, Laporan Tugas Akhir Sarjana, Universitas Sultan Syarif Kasim, Riau, 2011.

Memahami Model Enkripsi dan Security Data, Halaman 43, Andi, Yogyakarta.

Michael Siregar, Ivan, *Membongkar Source Code berbagai Aplikasi Android*, Halaman 227, Gava Media, Yogyakarta.

Munir, Rinaldi, *Kriptografi*, Halaman 301, Informatika, Bandung, 2006.

Putra, Darma, *Pengolahan Citra Digital*, Halaman 71, Andi, Yogyakarta, 2009.

Rahmanto, Arif, *Enkripsi Sms (Short Message Service) Dengan Menggunakan Algoritma Rc6 Pada Sistem Operasi Android*, Laporan Tugas Akhir Sarjana, Universitas Sultan Syarif Kasim, Riau, 2012.

Safaat Hasibuan, Nazruddin, *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*, Informatika, Bandung, 2011

Suprapti, Iswanti, *Studi Sistem Keamanan Data Dengan Metode Public Key Cryptography*, Laporan Tugas Akhir, Institut Teknologi Bandung, Bandung, 2003.

_____.Teknologi Penyadapan Di Indonesia. [Online] available. http://id.shvoong.com/internet-and-technologies/software/2277394 teknologi penyadapan-di-indonesia/#ixzz26pY0jVbf , Diakses 15 Mei 2012

_____.Berbagai peraturan hukum yang terkait dengan penyadapan di Indonesia semrawut dan tumpang tindih satu sama lain. Karena banyaknya otoritas yang memberikan izin untuk penyadapan. [Online] available. http://nilah.com, Diakses 30 September 2012

_____. *Binary Truth Table*. Dari: http://www.tutorvista.com/math/binary-truth-table, Diakses 19 Juni 2013

_____. *ASCII Table.* [Online] available. http://ascii-table.com/conversions.php, Diakses 31 Mei 2013

_____. *ASCII Table*. [Online] available. http://www.aubraux.com/design/ascii-table.php, Diakses 31 Mei 2013.

_____. *COMPUTERS Measurements for Memory & Storage*. [Online] available. http://www.athropolis.com/popup/c-comp2.htm, Diakses 19 Agustus 2013.